Post-prediction Inference

Siruo Wang[1], Tyler H. McCormick[2], and Jeffrey T. Leek[1]

[1]Department of Biostatistics, Johns Hopkins University

[2]Departments of Statistics & Sociology, University of Washington

January 21, 2020

Abstract

Many modern problems in medicine and public health leverage machine learning methods to predict outcomes based on observable covariates [1, 2, 3, 4]. In an increasingly wide array of settings, these predicted outcomes are used in subsequent statistical analysis, often without accounting for the distinction between observed and predicted outcomes [1, 5, 6, 7, 8, 9]. We call inference with predicted outcomes *post-prediction inference*. In this paper, we develop methods for correcting statistical inference using outcomes predicted with an arbitrary machine learning method. Rather than trying to derive the correction from the first principles for each machine learning tool, we make the observation that there is typically a low-dimensional and easily modeled representation of the relationship between the observed and predicted outcomes. We build an approach for the *post-prediction inference* that naturally fits into the standard machine learning framework. We estimate the relationship between the observed and predicted outcomes on the testing set and use that model to correct inference on the validation set and subsequent statistical models. We show our *postpi* approach can correct bias and improve variance estimation (and thus subsequent statistical inference) with predicted outcome data. To show the broad range of applicability of our approach, we show *postpi* can improve inference in two totally distinct fields: modeling predicted phenotypes in re-purposed gene expression data [10] and modeling predicted causes of death in verbal autopsy data [11]. We have made our method available through an open-source R package: [`https://github.com/SiruoWang/postpi`]

1 Introduction

The past decade has seen both an explosion in data available for precision health [12, 13, 14] and, simultaneously, user-friendly tools such as the caret package [15] and Scikit-learn [16] that make implementing complex statistical and machine learning methods possible for an increasingly wide range of scientists. For example, machine learning from electronic medical records is used to predict phenotypes [1, 17], genomic data is used to predict health outcomes [2], survey data is used to predict cause of death in settings where deaths happen outside of hospitals [3, 18]. The increased focus on ideas like precision medicine means the role of machine learning in medicine and public health will only increase [4]. As machine learning plays an increasingly critical role across scientific disciplines, it is critical to consider all sources of potential variability in downstream inference to ensure stable statistical results [19].

In many settings, predicted outcomes from machine learning models become inputs into subsequent statistical analyses. One example from genetics is association studies between genetic variants and Alzheimer's disease for young adults. Because young adults have not developed Alzheimer's disease, it is difficult to associate the phenotype with genetic variants. However, these adults' older relatives can be used to predict the ultimate phenotype of participants in the study using known inheritance patterns for the disease. The predicted outcome can be used in place of the observed Alzheimer's status when performing a genome-wide association study [6].

This is just one example of the phenomenon of *post-prediction inference* or *posti* for short. Researchers predict an outcome or phenotype they care about and then use that predicted outcome in subsequent downstream models. Though common, this approach poses multiple potential statistical challenges. The predicted outcomes may be biased, may have less variability than the actual outcomes, or the predictions may be based on the same covariates that would be used as independent variables in the subsequent statistical analyses leading to overfitting [20]. Currently, the standard approach to modeling predicted outcomes is to treat them as if they were the observed data [1, 5, 6, 7, 8, 9]. This may lead to biased or overly optimistic statistical inference.

Here we focus on developing theoretical and simulation-based corrections for statistical models using predicted outcomes. We focus specifically on settings where a predicted outcome becomes the outcome (dependent) variable in subsequent analysis. We build our approach into the common structure for machine learning problems of dividing the observed data into training, testing, and validation set. We can build the prediction model on the training set, then estimate the relationship

between the predicted and observed outcomes on the testing set, and use this estimated relationship to evaluate statistical inference using predicted outcomes on the validation set. An advantage of this approach is that it is not specific to a particular machine learning model. That is, we do not need to know *a priori* the expected out of sample performance for a given method. Instead, we assume that the relationship between the predicted and observed outcomes on the testing set well-characterizes the same relationship on the validation set.

The setting we describe has parallels with multiple imputation [21] for missing data, but has several distinct features. Any prediction problem could be cast as a missing data problem where all of the values are missing and no missingness mechanism distinguishes observed and unobserved outcomes. The reason is that on the validation set or the subsequent analyses in practical problems there are no observed outcome data. Multiple imputation also frequently relies on a generative model for simulating data, however in our setting, we wish to build a framework that can be used for any machine learning model, regardless of its operating characteristics. We, therefore, need a new methodology that can use a black-box machine learning algorithm but build a simple model for the relationship between the predicted and observed outcome data. This problem is related to the idea of errors-in-variables [22] or measurement error models [23], where either the outcome or the covariates are measured with error. However, in prediction problems, we can no longer assume that the errors are independent of the predicted values, since the machine learning predictions may be more accurate for subsets of the y values.

Aside from its utility in medicine and public health, the methods we propose are also broadly applicable in the social sciences. In political science, for example, machine learning tools classify sentiment or political identification in segments of text and then fit regression models to identify features of text leaning towards one party or another[24]. In urban sociology, researchers used machine learning tools to infer the race of household heads subject to eviction, then used regression models to understand heterogeneity in circumstances related to evictions of individuals of a particular race[25].

We apply our *postpi* approach to two open problems: modeling the relationship between gene expression levels and tissue types [2], and understanding trends in (predicted) cause of death [26, 27]. We show that our method can reduce bias, appropriately model variability, and correct hypothesis testing in the case where only the predicted outcomes are observed. We also discuss the sensitivity of our approach to changes in the study population that might lead to a violation of the assumptions of our approach. Our *postpi* approach is available as an open-source R package available from Github:

[https://github.com/SiruoWang/postpi].

The remainder of the paper is organized as follows. In the remainder of this section, we provide an example of the setting where our approach would be valuable. Then, Section 2 we present our results, followed by evaluation in Section 3. We conclude in Section 4.

1.1 Example problem

Consider an example, where we have observations for the outcome y_i and covariates x_{ij} for $i = 1, ..., n$,$j = 1, ..., p$. We use X_i to denote vector $[x_{i1}, ..., x_{ip}]$. In this example we will assume there are three separate models for the data. The first is the true state of the relationship between y and x which we will denote by $h(\cdot)$:

$$y_i = h(X_i) + e_{TSi} \tag{1}$$

The second is a prediction model using any arbitrary machine learning method. This model makes predictions of the outcome conditional on the covariates and may or may not accurately reflect the true relationship between y and x. Here $f(\cdot)$ denotes the prediction model and y_p denotes the predicted outcome:

$$y_{pi} = f(X_i) \tag{2}$$

The third is the inferential model that we will subsequently apply to the data for scientific interpretation. For example, it is common to use linear regression to relate a continuous outcome to a set of covariates. Here we use M_{x_i} to denote covariate matrix $\begin{bmatrix} 1 & x_{i1} & ... & x_{ip} \end{bmatrix}^T$ and this regression model for continuous data could be of the form:

$$y_i = M_{x_i}\vec{\beta} + e_{INF_i} \tag{3}$$

We could imagine fitting Equation 3 using either the observed outcome y_i or the predicted outcome y_{pi}. Standard statistical inference procedures assume that the y_i are observed without error. However, when we use predicted outcomes y_{pi} then Equation 3 no long appropriately reflects our uncertainty about the outcome. Figure 1 shows a simple simulated example of this idea. We simulate covariates x_{i1}, x_{i2}, x_{i3}, x_{i4} and error terms e_{TSi} from normal distributions and simulate the observed outcome y_i again from a normal distribution with mean function a linear combination

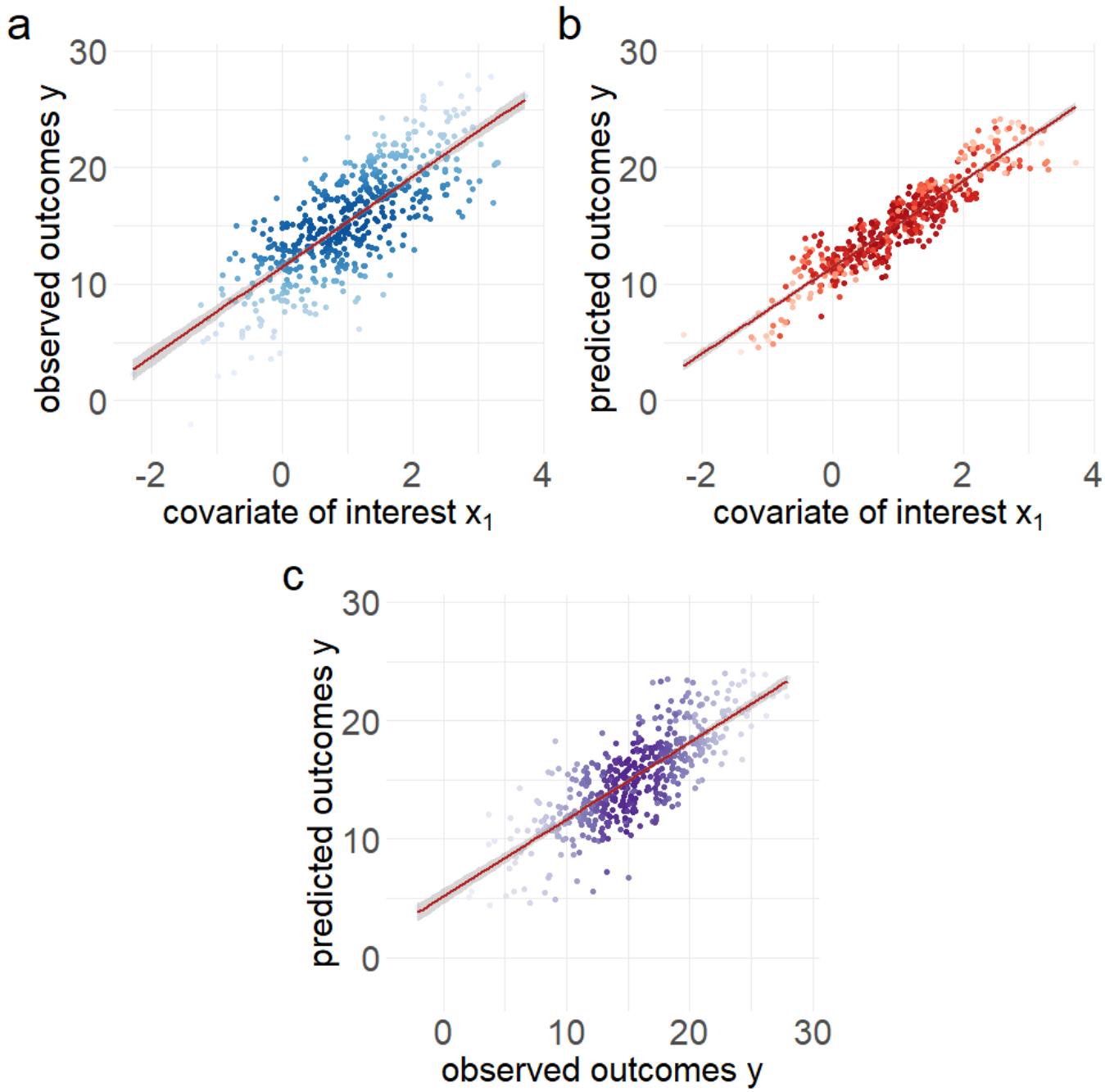

Figure 1: **Simulated example.** Data were simulated from the ground truth model as a linear model. (a) Observed outcomes versus the covariate of interest. The x-axis shows the covariate of interest x_1 and the y-axis shows the observed outcomes of y. (b) Predicted outcomes versus the covariate of interest. The x-axis shows the covariate of interest x_1 and the y-axis shows the predicted outcomes of y_p. (c) Observed outcomes versus predicted outcomes. The x-axis shows the observed outcomes of y and the y-axis shows the predicted outcomes of y_p.

of covariates x_{i1}, x_{i2}, x_{i3}, x_{i4}. Then we separate the simulated values into training, testing, and validation set. On the training set, we train a random forests [28, 29] model using all covariates x_{i1}, x_{i2}, x_{i3}, x_{i4} and observed outcome y_i. Then we apply this machine learning model to the observed covariates x_{i1}, x_{i2}, x_{i3}, x_{i4} on the testing set to obtain the predicted values y_{pi}. Now we estimate the relationship between the predicted and observed outcomes on the testing set.

In the first panel Figure 1(a) we illustrate the true relationship between the simulated y and x_1 (blue color). In the second panel Figure 1(b) we show the predicted values y_p versus x_1 (red color). You can see that the relationship has changed, with different slope and variance. In the third panel Figure 1(c) we show the relationship between the observed and predicted outcomes. In this simulated example, we know that the estimated coefficient for the relationship between the observed outcome y and x is 3.87 with a standard error of 0.14. However, when we fit model using the predicted outcome y_p we get an estimate of 3.7 with a standard error of 0.068. This example reflects that there is bias existing and standard error is reduced after using predicted outcome y_p in place of observed y.

To adjust for error in predictions, one option would be to derive bias and standard error corrections for a specific machine learning method. This approach would leverage knowledge about how a specific prediction tool works. To compute the bias and standard errors analytically we both (a) need to know what machine learning model was used and (b) need to be able to theoretically characterize the properties of that machine learning model's predictions. This approach, however, would then restrict an analyst to only one machine learning approach (i.e. the one with properties that have been worked out). As we observe in Figure 1(c), however, the relationship between the observed and predicted outcome can easily be modeled for a variety of machine learning techniques. We leverage this observation in the subsequent approach.

The key idea of our approach is we use this relationship and the data on the testing set, to estimate the bias and variance introduced by using predicted outcomes. This approach does not require idiosyncratic information about each machine learning approach and, instead, assumes that a relatively simple model captures the relationship between the predicted and observed outcomes.

2 Method

2.1 Overview of our approach

Our goal is to develop a method for correcting inference in statistical models using predicted outcomes. To do this we make the following assumptions about the structure of the data and model. We assume that the data are generated from an unknown data generating model of the form:

$$g\left(E[y_i|x_i, \boldsymbol{z_i}]\right) = h(x_i, \boldsymbol{z_i}). \tag{4}$$

Here x_i denotes the covariate of interest and z_i denotes other covariates. This model represents the "true state of nature" but is not directly observed in any practical problem.

We also assume that in a new data set it may be too expensive, too time-consuming, or too difficult to collect outcome variable y_i for all samples. We, therefore, will attempt to predict this outcome with an arbitrary machine learning algorithm $f(\cdot)$ so that $y_{pi} = f(x_i, z_i)$ is the predicted outcome based on the observed covariates.

However, the primary goal of our analysis is not to simply predict outcomes but to relate them directly to a subset of covariate x_i (i.e. same covariates used to make predictions in the new samples) or to a new set of x_i (i.e. new covariate collected only in the new samples). Since the data generating distribution is unknown, we fit a generalized linear regression model to relate outcomes (observed or predicted) to covariates. Here, M_{x_i} is the matrix notation of the covariate of interest:

$$g\left(E[y_i|x_i]\right) = M_{x_i}\vec{\beta}. \tag{5}$$

When outcome is observed, we can directly compute the estimate of $\vec{\beta}$. However, we assume that in a new sample it will not be possible to observe outcome, so predicted outcome y_{pi} will be used in Equation 5.

The most direct approach to performing post prediction inference is to use predicted outcomes and ignore the fact that they are predicted. This approach can lead to bias and reduced variance for estimated coefficients as we saw in the simple example in Section 1.1. We will demonstrate that this approach produces consistently inaccurate inference in the simulation and real application settings. Despite these potential biases, this approach to direct use of predicted outcomes in inferential models is popular in genomics [9], genetic [6], public health [18], and EHR phenotyping [1] among other applications.

Another strategy would be to try to directly derive the properties of the coefficients and standard errors in the subsequent inference model using the definition of the machine learning algorithm $f(\cdot)$. When a prediction is based on a simple regression model, this may be possible to do directly. However, machine learning models now commonly include complicated algorithmic approaches involving thousands or millions of parameters, including k-nearest neighbors [30], SVM [31], random forests [28, 29] and deep neural networks [32].

We instead focus on modeling the relationship between the observed and predicted outcomes. Our key insight is that even when we use a complicated machine learning tool to predict outcomes, a relatively simple model can describe the relationship between the observed and predicted outcomes (Figure 2). We then use this estimated relationship to compute bias and standard error corrections for the subsequent inferential analyses using predicted values as the outcome variable.

Based on the observation in Figure 2, we relate the observed to the predicted data through a flexible model $k(\cdot)$:

$$y_{pi} = k(y_i) \tag{6}$$

For continuous outcomes, we can estimate the relationship as a linear regression model. For categorical outcomes, we can use a logistic regression model or a simple machine learning model to estimate the relationship between the observed and predicted outcomes. To fit this relationship model we take advantage of the standard structure of machine learning model development. The observed data is split into training, testing, and validation set. We can build the prediction model on the training set and then compute an unbiased estimate of the relationship model on the testing set. Using this relationship model we derive a correction for the estimates, standard errors, and test statistics for our inference model. Then on the validation set, we can evaluate the quality of our correction in an independent sample.

In Section 2.2, we derive an analytic correction for the case where (1) the outcome is continuous, based on the assumed model structure, and (2) the covariate of interest in the subsequent inference model performed on predicted data is a subset of covariates that used to make such predictions. This analytic correction holds regardless of the choice of machine learning algorithm $f(\cdot)$ used to make the predictions, provided that there is a specific relationship between the observed and predicted outcomes. In Section 2.3, we generalize this approach using a derived bootstrap procedure for two reasons. First, it no longer assumes that predicted outcomes are continuous. Second, the covariate of interest in the subsequent inferential analyses can be a new variable collected only in a new

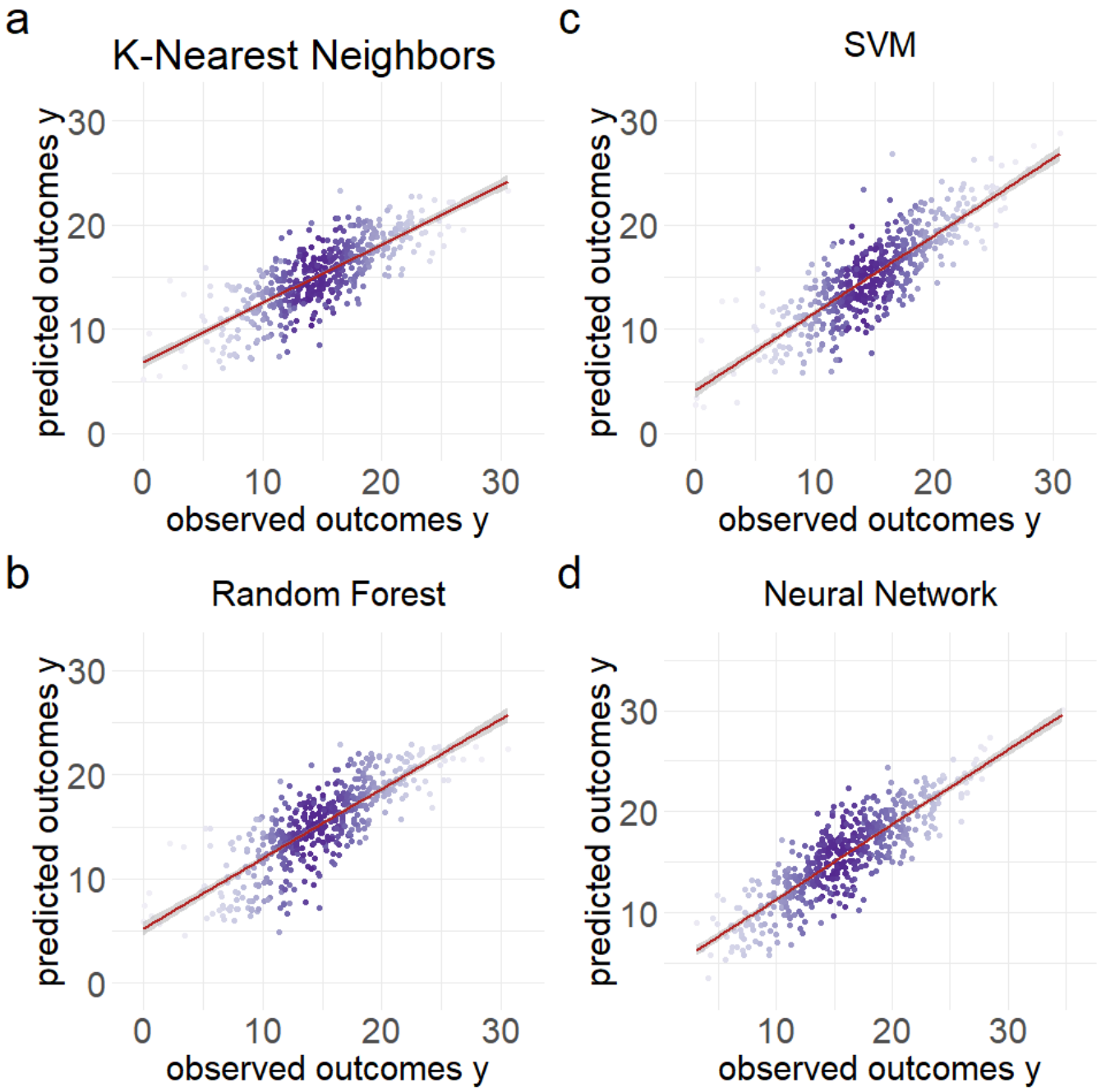

Figure 2: **Relationship between the observed and predicted outcomes using different machine learning models.** Data were simulated from the ground truth model as a linear model with normally distributed noise. On the x-axis is the observed outcome of y and on the y-axis are the predicted outcomes y_p. We show that regardless of the prediction method (a) k-nearest neighbors, (b) Random Forests, (c) SVM, or (d) Neural Network, the observed and predicted outcomes follow a distribution that can be accurately approximated with a regression model.

sample.

2.2 Derivation correction

For continuous outcome y_i, we assume that the predicted value y_{pi} follows a normal distribution centered around the observed value y_i (see Equation 7), and we consider a relationship model such that the mean value of y_{pi} is modeled as a linear function of y_i: $E[y_{pi}|y_i] = M_{y_i}\vec{\gamma}$. Recall the inference model we assume in Equation 5 , in the case of continuous outcomes, observed outcome y_i is modeled as a normal distribution with mean $M_{x_i}\vec{\beta}$ and variance σ^2_{INF} (see Equation 8). On the testing set, we consider the following models:

$$y_{pi} \mid y_i \sim \mathcal{N}(\gamma_0 + \gamma_1 y_i, \sigma^2_{REL}) \tag{7}$$

$$y_i \mid x_i \sim \mathcal{N}(M_{x_i}\vec{\beta}, \sigma^2_{INF}) \tag{8}$$

From the relationship model in Equation 7, we find the maximum likelihood estimation (MLE) for parameters γ_0, γ_1, and σ^2_{REL}. We use $\hat{\gamma_0}$, $\hat{\gamma_1}$, $\hat{\sigma^2_{REL}}$ for MLE notation which are obtained by finding the parameter values that maximize the likelihood function. That is, $\hat{\gamma}_0 = \frac{1}{n}\sum_{i=1}^n y_{pi} - \hat{\gamma}_1 \frac{1}{n}\sum_{i=1}^n y_i$, $\hat{\gamma}_1 = \frac{\sum_{i=1}^n (y_i - \overline{y})(y_{pi} - \overline{y_p})}{\sum_{i=1}^n (y_i - \overline{y})^2}$ and $\hat{\sigma}^2_{REL} = \frac{1}{n}\sum_{i=1}^n (y_{pi} - \hat{\gamma}_0 - \hat{\gamma}_1 y_i)^2$ where $\overline{y}$ is the mean for observed outcomes y_i and $\overline{y_p}$ is the mean for predicted outcomes y_{pi}. Similarly, from the inference model in Equation 8, we find the maximum likelihood estimator (MLE) for parameter σ^2_{INF}, denoted as $\hat{\sigma}^2_{INF}$. That is, $\hat{\sigma}^2_{INF} = \frac{1}{n}\sum_{i=1}^n (y_i - M_{x_i}\hat{\beta})^2$.

The usual strategy is to perform inference relating the predicted outcomes y_p and the covariate of interest x (i.e. interchange y_i with y_{pi} in Equation 5) [1, 5, 6, 7, 8, 9]. To correct inference for $\hat{\beta}_p$ performed on data with predicted outcome y_p, we first need to find the conditional expectation and variance of predicted outcome given covariates. This expectation can be written as:

$$\begin{aligned} \mathbf{E}\left[y_p|x\right] &= \mathbf{E}\left[\mathbf{E}\left[y_p|x,y\right]|x\right] \\ &\approx \mathbf{E}\left[\mathbf{E}\left[y_p|y\right]|x\right] \\ &= \gamma_0 + \gamma_1 M_x \vec{\beta}. \end{aligned} \tag{9}$$

In the second step of Equation 9, we have made the approximation of using the relationship between y_p and y to model the conditional expectation $E[y_p|x,y]$ (see supplement Section 1 for full derivation). Using the iterated expectation above alleviates the need to define the expectation of y_p given x directly. The reason we make such approximation is that the conditional expectation of y_p

given x may be arbitrarily complicated if we use a method such as random forests [28, 29], neural networks [32], or boosted trees [33]. Instead, we only need to compute $\mathbf{E}[y_p|y]$ for approximation, which is directly estimable from the data.

For inference to relate y_p and x through a regression model, we also must compute the variance of the estimate $\hat{\beta}_p$. The challenge is that the variance is not simply calculated by fitting the regression model between y_p and x. Instead, we derive the variance of the predicted outcome from fitting both the relationship model between the predicted and observed outcomes (y_p and y) and the inference model between the observed outcomes and covariate of interest (y and x). This variance can be written as:

$$\begin{aligned}\mathbf{Var}[y_p \mid x] &= \mathbf{E}[\mathbf{Var}[y_p \mid x, y] \mid x] + \mathbf{Var}[\mathbf{E}[y_p \mid x, y] \mid x] \\ &\approx \mathbf{E}[\mathbf{Var}[y_p \mid y] \mid x] + \mathbf{Var}[\mathbf{E}[y_p \mid y] \mid x] \\ &= \sigma^2_{REL} + \gamma_1^2 \sigma^2_{INF}.\end{aligned} \tag{10}$$

In the second step of Equation 10, we again have made the approximation of using the relationship between y_p and y to model the conditional variance $Var[y_p|x, y]$ (see supplement Section 1 for full derivation). The conditional variance computed above does not consider bias correction. However, our procedure is set up to estimate and remove bias in the inference step due to prediction error. The bias in our estimate can be written as: $Bias_\beta(\hat{\beta}_p) = \mathbf{E}_{y|X}(\hat{\beta}_p) - \beta$. In general, the bias of $\hat{\beta}_p$ relative to β will not be zero due to errors introduced by the prediction model $f(\cdot)$ in Equation 2.

To correct this problem, we take advantage of the structure of a machine learning problem and use the testing set to estimate the bias. On the testing set, both y and y_p are observed, so we can make a direct comparison between the two coefficients. We estimate the bias as: $\Delta_{bias} = \hat{\beta}_p^{\,test} - \hat{\beta}^{test}$, where $\hat{\beta}_p^{\,test}$ is the no correction estimate performed on the predicted data and $\hat{\beta}^{test}$ is the estimate performed on the observed data.

We can fix Δ_{bias} estimated on the testing set and use it to correct the bias on the validation set due to prediction error: $\hat{\beta}^{der} = \hat{\beta}_p^{\,val} - \Delta_{bias}$. Here, $\hat{\beta}_p^{\,val}$ denotes the no correction estimate using the predicted data on the validation set. Our corrected estimate $\hat{\beta}^{der}$ is improved from no correction by incorporating both the fixed bias term and the estimated conditional variance of predicted outcome. Now the bias of coefficient $\hat{\beta}^{der}$ is approximately zero (see supplement Section 1 for full derivation) and the variance is adjusted to $(M_x^{val^T} M_x^{val})^{-1}(\hat{\sigma}^2_{REL}{}^{test} + \hat{\gamma_1}^{2test} \cdot \hat{\sigma}^2_{INF}{}^{test})$ (see supplement Section 1 for full derivation). Thus, we propose $\hat{\beta}^{der}$, with both bias and variance corrections, behaves more similar to the gold standard estimate $\hat{\beta}$ performed on the observed data.

Using the corrected estimate and standard error, we can now perform hypothesis test for the coefficient of interest in our regression model. Recall that the "inference model" in Equation 5 we are studying is of the form $y_i|x_i = M_{x_i}\vec{\beta} + \epsilon_i$ where we use y_p in place of y. To test for the null hypothesis against the alternative of the form: $H_0 : \beta_{p_k} = 0$ vs. $H_a : \beta_{p_k} \neq 0$ (β_{p_k} is the k-th component of the estimator vector), we can then build the test statistic:

$$t_{\hat{\beta_k}^{der}} = \frac{\hat{\beta_k}^{der} - \beta_{p_k}}{\hat{se}^{der}} = \frac{\hat{\beta_k}^{der}}{\sqrt{(M_x^{val^T} M_x^{val})^{-1} \cdot (\hat{\sigma}^2_{REL}{}^{test} + \hat{\gamma_1}^{2test} \cdot \hat{\sigma}^2_{INF}{}^{test})}}. \tag{11}$$

We define a decision rule to decide whether the null hypothesis shall be rejected or not. One way is to compare the test statistic. We reject the null hypothesis $H_0 : \beta_{p_k} = 0$ in favor of the alternative hypothesis $H_a : \beta_{p_k} \neq 0$ at the significance level α when $t_{\hat{\beta_k}^{der}} > t^{\alpha}_{n-p}$, where t^{α}_{n-p} is from the t statistical table with p degrees of freedom and significance level α.

2.3 Bootstrap simulation

In the previous Section 2.2, we concentrated on a setting where the outcome is continuous and approximately normally distributed. We also made assumptions that the covariate of interest in the subsequent inferential model comes from the set of covariates used to get predicted values y_p through a machine learning tool. In this section, we expand our scope by proposing a bootstrap approach for correcting the bias and variance in the downstream inferential analyses. This approach can be applied for continuous, non-normal data, categorical data, or count data. For our approach we make the following assumptions: (1) that the relationship between the observed and predicted outcome can be modeled through a specific simple model, (2) the relationship model will hold out of sample, and (3) we have a training, testing, and validation set for building the prediction model and estimating parameters of the relationship model.

The first step of our bootstrap procedure follows the standard process for machine learning by randomly splitting the data into a training, testing, and validation set. The algorithm then proceeds as follows:

Bootstrap Procedure:

1. Use the training set data to estimate the outcome prediction function $y_{pi} = \hat{f}(x_i)$
2. Use the testing set data to estimate the relationship model $y_i = k(y_{pi})$, where k can be any

flexible function.

3. Use the validation set to bootstrap as follows

for Bootstrap iteration b = 1 to B **do**

(i) For $i = 1, 2, ..., n$, sample predicted values and the matching covariates (y^b_{pi}, x^b_i) with replacement

(ii) Simulate values from the relationship model $\tilde{y}^b_i = k(y^b_{pi})$ using the function $k(\cdot)$ estimated from the testing set in Step 2

(iii) Fit the inference model $g\left(E[\tilde{y}^b_i|x^b_i]\right) = M^b_x\vec{\beta}$ using the simulated outcomes $\tilde{y}^b_i$ which build in the prediction error from the relationship model and the matching model matrix based on the sampled covariates x^b_i

(iv) Extract the coefficient estimator and its standard error from the inference model in (iii), denoted as $\hat{\beta}^b$ and $se(\hat{\beta}^b)$, and save them to the coefficient estimator list and the standard error list accordingly

end for

4. Estimate the coefficient and standard error using a median function: $\hat{\beta}^{boot} = median\left(\hat{\beta}^1, \hat{\beta}^2, ..., \hat{\beta}^B\right)$ and $\hat{se}^{boot} = median\left(\hat{se}\left(\hat{\beta}^1\right), \hat{se}\left(\hat{\beta}^2\right), ..., \hat{se}\left(\hat{\beta}^B\right)\right)$

The bootstrap approach builds in two types of errors: the error due to random sampling and the prediction error. The prediction error is introduced by sampling from the relationship model in the ***for*** loop Step 3(ii). We again make the simplifying assumption that y_i and y_{pi} can be related through a model that is easy to fit. We can focus here on the class of generalized linear models, but in the ***Bootstrap Procedure*** Step 2, the relationship function $k(\cdot)$ could be more general, even flexible as a machine learning algorithm, provided it can be easily estimated and sampled. The advantage of the relationship model is that we do not need to assume the type or complexity of the function $f(\cdot)$ used to make the predictions. It can be arbitrarily complicated so long as the estimated relationship between the observed and predicted values can be sampled.

3 Evaluation

Here we perform a series of comparisons between our derivation and bootstrap approaches to the alternative of no correction. No correction means that the outcome is predicted and then treated as

observed in downstream inference models. We can compare our methods *postpi derivation*, *postpi bootstrap* and no correction to the case where we have the observed outcome. In practical examples, we do not observe the true outcome, but for the purposes of these comparisons, we can either use simulated data or use data from the validation set where the observed outcome is known for comparison purposes.

3.1 Simulated data

We simulate the independent covariate x, the error term e_{TS}, and then simulate observed outcome y using the "true state of nature" model in Equation 4. The "true state of nature" is not directly observed in practical problems but can be specified in simulated problems. We consider both the case of a continuous outcome and a binary outcome.

3.1.1 Continuous case

For the continuous case we simulate covariates x_{ij} and error terms e_{TSi} from normal distributions, and simulate the observed outcome y_i using a linear function $h(\cdot)$ as the "true state of nature" model for $i = 1, ..., n,\ j = 1, ..., p$.

In each simulation cycle, we set the total sample size $n = 900$ and the dimension of covariate matrix $p = 4$. To mimic a complicated data generating distribution and make predictions sufficiently variable for illustration purposes, we generate data including both linear and smoothed terms. For the smoothed terms, we use Tukey's running median smoothing with a default smoothing parameters"3RS3R" [34]. The error terms are also simulated from a normal distribution with independent variance. The model specification is:

$$
\begin{aligned}
x_{i1}, x_{i2}, x_{i3} &\sim \mathcal{N}(1, 1) \\
x_{i4} &\sim \mathcal{N}(2, 1) \\
e_{TSi} &\sim \mathcal{N}(0, 1) \\
y_i &= \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 \cdot smooth\left(x_{i3}\right) + \beta_4 \cdot smooth\left(x_{i4}\right) + e_{TSi}
\end{aligned}
$$

We create a training, testing and validation set by randomly sampling the observed data into three equal size groups each with sample size 300. To mimic a more realistic setting, we assume that we are only interested in associating the outcome (y_i) and one covariate (in this case x_{i1}), and we will use a linear inference model to quantify this relationship.

For our simulation we use random forests [28, 29] to estimate the prediction function $f(\cdot)$. We fit the random forests model on the training set using all of the covariates $x_{i1}, x_{i2}, x_{i3}, x_{i4}$ as features to predict the observed outcomes y_i. This prediction, while possibly very accurate at predicting outcomes, is not designed to estimate the "true state of nature" or to perform statistical inference.

n testing set we apply the trained prediction model to get predicted outcomes y_{pi}. We estimate the relationship between the observed and predicted outcome (y_i and y_{pi}) as a simple linear regression model: $y_i|y_{pi} \sim \mathcal{N}(M_y\vec{\gamma}, \sigma^2_{REL})$. We then use standard maximum likelihood estimation to approximate the parameter estimates, $\hat{\gamma}$ and $\hat{\sigma}^2_{REL}$.

Our evaluation of the performance of different methods is done on an independent validation set. We will compare inference directly with the predicted outcome (no correction), *post-prediction inference* through postpi derivation and *post-prediction inference* through postpi bootstrap. Since the validation data includes the observed outcome y_i (reserved for results validation but not directly observed in practical settings), we can compare the results of each approach to what would happen if we observed the outcome. The baseline model we are comparing to fits the regression model $E[y_i|x_{i1}] = \beta_0 + x_{i1}\beta_1$ to the observed data on the validation set. We then estimate the coefficient, standard error, and t-statistic using standard maximum likelihood estimation.

To fit the three correction approaches we first perform the following steps. On the training set, we estimate the prediction function $\hat{f}(\cdot)$. We then predict the outcome on the testing and validation sets to produce outcome predictions $y_{pi} = \hat{f}(x_{i1}, x_{i2}, x_{i3}, x_{i4})$.

To fit the no correction approach, we perform a regression of the form: $E[y_{pi}|x_{i1}] = \beta_{p0} + x_{i1}\beta_{p1}$, treating the predicted outcome as if it was observed and calculate the coefficient, standard error, and t-statistic using maximum likelihood, ignoring the fact that the outcome is predicted.

To fit the postpi derivation approach, we estimate the coefficient by estimating the bias between the coefficients we get using the observed and predicted data on the testing set. We estimate the variance of this term using both the inference model and the relationship model on the testing set. We then apply these corrections to calculate the postpi derivation estimate $\hat{\beta}_1^{der}$, standard error $\hat{se}^{der}$, and t-statistics $t^{der} = \frac{\hat{\beta}_1^{der}}{\hat{se}^{der}}$ on the validation set.

To fit the postpi bootstrap approach, we follow the ***Bootstrap Procedure*** Step 1-4 in Section 2.3. In Step 1 we fit the random forests [28, 29] prediction model on the training set. In Step 2 we estimate the relationship model $y_i|y_{pi} \sim \mathcal{N}(M_y\vec{\gamma}, \sigma^2_{REL})$ on the testing set. In Step 3 we first set the bootstrap size $B = 100$ to start the ***for*** loop, and then repeat Step 3(i)-(iv) on the validation set. Specifically, in Step 3(ii) we estimate the relationship model $k(\cdot)$ as a linear function

and simulate values from the distribution: $\tilde{y}_i^b|y_{pi}^b \sim \mathcal{N}(M_{y_p}^b\hat{\gamma}, \hat{\sigma}_{REL}^2)$. Both the mean and standard deviation of the sampling distribution come from the estimated relationship model. In Step 3(iii) we fit a linear regression model as the inference model: $E[\tilde{y}_i^b|x_{i1}^b] = \beta_{p0} + x_{i1}^b\beta_{p1}$. We then estimate the postpi bootstrap coefficient $\hat{\beta}_1^{boot}$, standard error $\hat{se}^{boot}$, and t-statistic $t^{boot} = \frac{\hat{\beta}_1^{boot}}{\hat{se}^{boot}}$.

Across 500 simulated cases, we fix the values of $\beta_2 = 0.5, \beta_3 = 3, \beta_4 = 4$ and set β_1 to be a range of values in $[-6, -5, ..., 5, 6]$ for the covariate of interest x_{i1} in the downstream inferential model. We then compute the estimates, standard errors, and t-statistics for β_1 with no correction, postpi derivation, and postpi bootstrap approaches and compare them to the baseline results where the outcome is observed.

In this simulation example, the prediction has relatively little bias, so the estimated coefficients using the predicted outcome are relatively close to the estimates using the observed outcome in Figure 3(a). However, the standard errors for the no correction approach (red color) in Figure 3(b) is much lower than what we would have observed in the observed outcomes. This is because the prediction function attempts to capture the mean function, but not the variance in the observed outcome. We compute the root mean squared error (RMSE) [35] to show that both the postpi derivation and postpi bootstrap approaches outperform the no correction approach. The standard errors are closer to the truth with an RMSE reduced from 0.087 for no correction (red color) to 0.073 for postpi derivation (green color), and further improved to 0.020 for postpi bootstrap (blue color) in Figure 3(b). The improved standard errors are reflected in improved t-statistics using the postpi derivation and postpi derivation bootstrap approaches in Figure 3(c), with RMSE reduced from 20.80 for no correction (red color) to 12.14 for postpi derivation (green color) and further to 3.84 for postpi bootstrap (blue color).

3.1.2 Binary case

For the binary case we simulate a categorical covariate x_{ic}, continuous covariates x_{i1}, x_{i2}, and an error term e_{TSi}, and then simulate the observed outcome y_i assuming a generalized linear model $h(\cdot)$ for $i = 1, ..., n$. In this case, we specify the "true state of nature" model $h(\cdot)$ to be a logistic regression model. To simulate observed outcomes y_i, we first set up covariates through a linear combination where we smooth a subset of continuous covariates using Tukey running median smoothing [34] and include errors to increase variability in outcomes y_i. We apply the inverse logit function to the linear predictor to simulate probabilities which we use to simulate Bernoulli outcomes ($y_i = 0$ or 1)

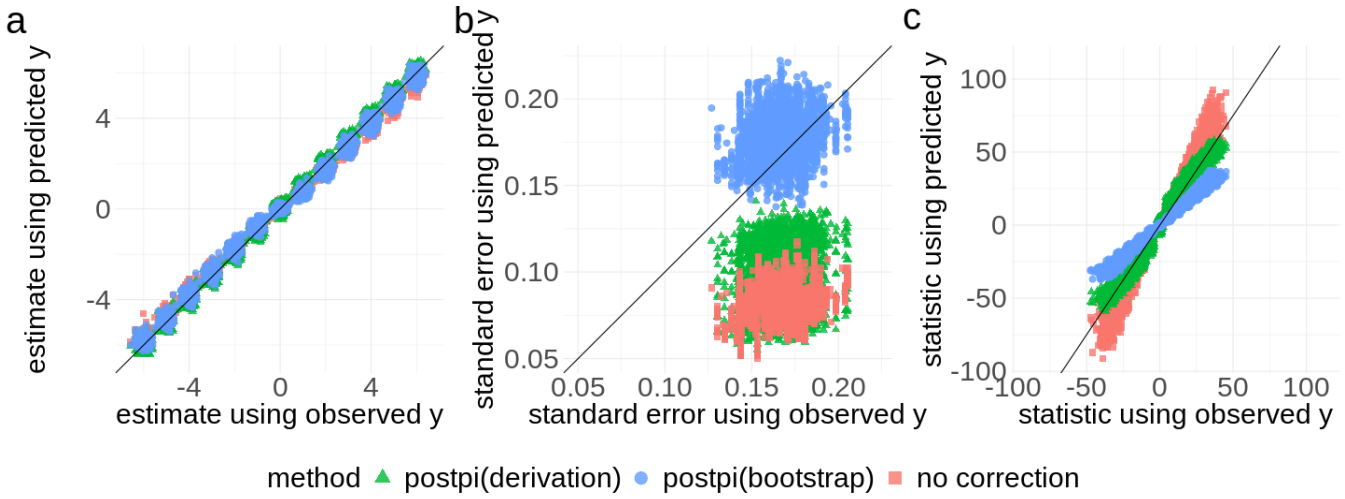

Figure 3: **Continuous simulation.** Data were simulated from the ground truth model as described in Section 3.1.1. On the x-axis are the values calculated using the observed outcome and on the y-axis are the values calculated using no correction (red color), postpi derivation (green color) and postpi bootstrap (blue color). We show (a) the estimates, (b) the standard errors and (c) the t-statistics. The postpi bootstrap and postpi derivation approaches improve the standard errors and t-statistics compared to no correction.

through binomial distributions. We simulate as follows:

$$
\begin{aligned}
x_{i1} &\sim \mathcal{N}(1,1) \\
x_{i2} &\sim \mathcal{N}(2,1) \\
x_{ic} &\sim Multinom(1,(1/3,1/3,1/3)) \\
e_{TSi} &\sim \mathcal{N}(0,1) \\
z_i &= \beta_B 1(x_{ic}=B) + \beta_C 1(x_{ic}=C) + \beta_1 \cdot smooth\,(x_{i1}) + \beta_2 \cdot smooth\,(x_{i2}) + e_{TSi} \\
pr_i &= \frac{1}{1+e^{-z_i}} \\
y_i &\sim Binom(1, pr_i)
\end{aligned}
$$

We generate 1,500 samples for each iteration and and separate the data into a training, testing and validation set of equal size. We set $1(x_c = C)$ as the covariate of interest in the subsequent inferential model and compute the corrected coefficient estimate, standard error and test statistic using the ***Bootstrap Procedure*** Step 1-4 in Section 2.3.

We again use random forests [28, 29] as a machine learning tool and all independent covariates x_{ic}, x_{i1}, x_{i2} as features to estimate the prediction function $f(\cdot)$ on the training set. Then we apply

the trained prediction model on the testing and validation sets to get the predicted outcome y_{pi} as well as the probability pr_i of the predicted outcomes (i.e. $pr_i = Pr(y_i = 1)$). On the testing set, we use a logistic regression to estimate the relationship between the observed outcome and the predicted probability: $g(E[y_i = 1|pr_i]) = \gamma_0 + pr_i\gamma_1$, where $g(\cdot)$ is the natural log of the odds such that $g(p) = Ln(\frac{p}{1-p})$. Here we form the relationship model with the predicted probability. The reason is that the outcome is dichotomous, so we have little flexibility to model the variance in the observed outcome as a function of the predicted outcome. Instead, using predicted probability provides more flexibility to model the relationship.

In the case of a categorical outcome, the derivation approach no longer applies, so we apply the bootstrap correction only. On the validation, set we follow the ***Bootstrap Procedure*** Step 1-4. First we set the bootstrap size $B = 100$ to start the ***for*** loop. In Step 3(ii) $\tilde{y}_i^b = k(pr_i^b)$, we simulate values in two steps: (1) use pr_i^b and the estimated relationship model to predict the probability of getting the "success" outcome (i.e. $Pr(\tilde{y}_i^b = 1)$), and then (2) sample $\tilde{y}_i^b$ from a binomial distribution with the probability parameter as $Pr(\tilde{y}_i^b = 1)$ obtained from (1). In Step 3(iii) we again fit a logistic regression model as the inference model: $\mathrm{g}(E[\tilde{y}_i^b|x_c^b]) = \beta_{p0} + 1(x_c = C)^b\beta_{pC}$. Then we follow the postpi bootstrap algorithm to estimate the coefficient $\hat{\beta}^{boot}$, standard error $\hat{se}^{boot}$, and t-statistic $t^{boot} = \frac{\hat{\beta}^{boot}}{\hat{se}^{boot}}$.

In each simulation cycle described above, we fix the values of $\beta_1 = 1, \beta_2 = -2, \beta_B = 1$. Here we choose $1(x_c = C)$ as the covariate of interest in the downstream inferential analyses, so we set β_C to be a range of values in $[-2, -1.5, ..., 4.5, 5]$ for illustration purpose. In this example, we see bias in the coefficient estimate using the no correction approach (red color) in Figure 4(a) with RMSE 8.27 compared to the truth. This bias is corrected through the postpi bootstrap approach (blue color) in Figure 4(a) with RMSE reduced to 0.65. In this case, we see that both the estimates and standard errors are inflated in the case of no correction (red color) in Figure 4(a) and (b). Especially the standard errors for no correction (red color) in Figure 4(b) have extremely large RMSE 473.53 (large standard errors are not shown in the graph for clarity). This is due to the problem of sparsity in the dichotomous covariates under many simulations. By randomly bootstrapping samples in the ***Bootstrap Procedure*** Step 3(i) and using median function in Step 4, we can get robust standard errors. Therefore, the standard errors are underestimated with the postpi bootstrap approach (blue color) in Figure 4(b) and RMSE reduces to 0.041. The t-statistics are biased with no correction (red color) and this bias is reduced from RMSE 4.02 to 2.14 for postpi bootstrap approach (blue color) in Figure 4(c). We also observe a horizontal line (red color) in the t-statistic plot. This is

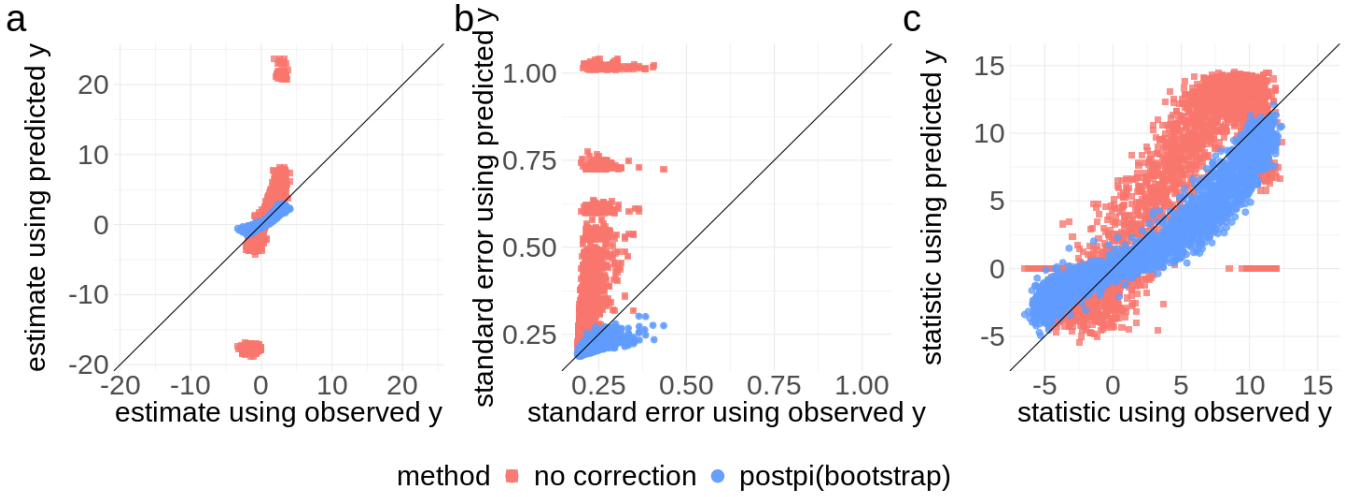

Figure 4: **Categorical simulation.** Data were simulated from the ground truth model as described in Section 3.1.2. On the x-axis are the values calculated using the observed outcome and on the y-axis are the values calculated using no correction (red color) and postpi bootstrap (blue color). We show (a) the estimates, (b) the standard errors and (c) the t-statistics. The Postpi bootstrap improves the estimates, leads to smaller standard errors, and less biased t-statistics compared to no correction. Extreme values of standard errors are also removed with the postpi bootstrap, which removes the t-statistics forced to zero under no correction.

due to large standard errors forcing the t-statistics close to zero in the no correction approach. The postpi bootstrap approach (blue color) removes these values along the horizontal line.

3.2 Applications

To demonstrate the wide applicability of our methodology for performing *post-prediction inference*, we present two examples from widely different fields: genomics and verbal autopsy analysis. These applications share very little in common scientifically, but represent two high profile examples where inference is typically performed with uncorrected predictions as the outcome (dependent) variable.

First, consider the "Recount2" Project (`https://jhubiostatistics.shinyapps.io/recount`) [36] which consists of RNA sequencing (RNA-seq) gene expression data for over 70,000 human samples aligned using a common pipeline processed in Rail-RNA [37]. While "Recount2" human samples have available gene expression information, not all samples contain observed phenotype information since the majority of the samples are pulled directly from public data on the sequence read archive [38]. However, we previously showed that many of these missing phenotype data can

be predicted from the genomic measurements [2]. Our goal is to perform inference using these predicted phenotypes.

Second, we describe the distribution of (predicted) causes of death. In regions of the world where routine monitoring of births and deaths is not possible, one approach to estimating the distribution of deaths by cause is the verbal autopsy (VA) survey. These surveys take place with a caregiver or relative of the decedent and ask about the circumstances surrounding the person's death, and typically take place when deaths happen outside of hospitals or routine medical care. Either expert guidance about the relationship between reported symptoms prior to death and the eventual cause or small "gold standard" datasets are used to train algorithms that predict causes of death based on reported symptoms. Algorithm development to predict causes of death is an active area of research and is challenging since data typically contain a mixture of binary, continuous, and categorical symptoms and many causes of death have similar presentations. After assigning a predicted cause of death, a common task is to describe patterns in the cause of death distribution. A scientist may be interested, for example, in how the distribution of deaths varies by region or by sex.

3.2.1 Predicting tissue types

We consider a motivating problem from the "Recount2" Project [36] (`https://jhubiostatistics.shinyapps.io/recount/`). In this example, the phenotype we care about is the tissue type where the RNA is sampled from. Understanding gene expression levels across tissues and cell types have many applications in basic molecular biology. Many research topics concentrate on finding which genes are expressed in which tissues, aiming to expand our fundamental understanding of the origins of complex traits and diseases [39, 40, 41, 42, 43]. The Genotype-Tissue Expression (GTEx) project [44], for example, studies how gene expression levels are varied across individuals and diverse tissues of the human body for a wide variety of primary tissues and cell types [39, 44]. Therefore, to better understand the cellular process in human biology, it is important to study the variations in gene expression levels across tissue types.

Even though tissue types are available in GTEx [44], they are not available for most samples in the "Recount2". In a previous paper [2], we developed a method to predict for those missing phenotypes using gene expression data. In this example, we collected a subset of samples that we have observed tissue types as breast or adipose tissues. We also had predicted values for the above samples calculated in a previous training set [2] using the 2281 expressed regions [10] as predictors.

Our goal in this example is to understand which of these regions are most associated with breast tissue in new samples (i.e. samples without observed tissue types) so that we can understand which measured genes are most impacted by the biological differences between breast and adipose tissues. Although here the phenotype we care about is the tissue types, especially breast and adipose tissues, our method can be broadly applied to any predictions to all phenotypes.

To test our method, we collected 288 samples from the "Recount2" with both observed and predicted tissue types. Among the observed tissue types, 204 samples are observed as adipose tissues and 84 samples are observed as breast tissues. The predicted values obtained from a previously trained data set [2] include the predicted tissue type (i.e. adipose tissue or breast tissue) and the probability for assigning a predicted tissue type. In this example, we compare no correction and postpi bootstrap approaches only since the outcomes we care about - tissue types are categorical.

The inference model we are interested in is: $g(E[y_i = 1|ER_i^j]) = \beta_0^j + \beta_1^j ER_j$. Here $g(\cdot)$ is the logit link function for $j = 1, \ldots, 2281$ (expressed regions) and $i = 1, \ldots, n$, n is the total number of samples in the "Recount2". In the model, $y_i = 1$ or $y_i = 0$ represents whether breast tissue is observed or adipose tissue is observed at the ith sample, and ER_i^j is the gene expression level for the jth region on the ith sample.

For this dataset (288 samples), we have binary tissue type outcomes. Since the predicted outcomes were obtained in a previously trained set [2], we only need to separate our data into a testing and validation set, each with a sample size $n = 144$. On the testing set, we fit a k-nearest neighbors [30] model to estimate the relationship between the observed tissue type and the probability of assigning the predicted value. On the validation set, we follow the ***Bootstrap Procedure*** in Section 2.3. Particularly in Step 3(ii), we simulate values from a distribution $\tilde{y}_i^b | pr_i^b \sim F_{\hat{\gamma}}$. Similar to we did with the simulated data in Section 3.1, in this example, we set F_γ to be a binomial distribution with the probability parameter (i.e. probability of assigning the outcome as breast cancer) estimated from the relationship model. In this way, we utilize the estimated relationship to account for necessary variations in simulated outcomes.

Among the 2281 expressed regions [10] used to make tissue type predictions [2], we care about the regions that have expression values across a relatively large amount of samples on the validation set. It is a well-known phenomenon that many RNA-seq measurements may be zero if the number of collected reads is low. To avoid highly variable model fits due to zero variance covariates, we only fit logistic regressions inference models to each filtered expressed region with expressed values over at least 20% samples. Under this filtering procedure, we include 101 expressed regions as regression

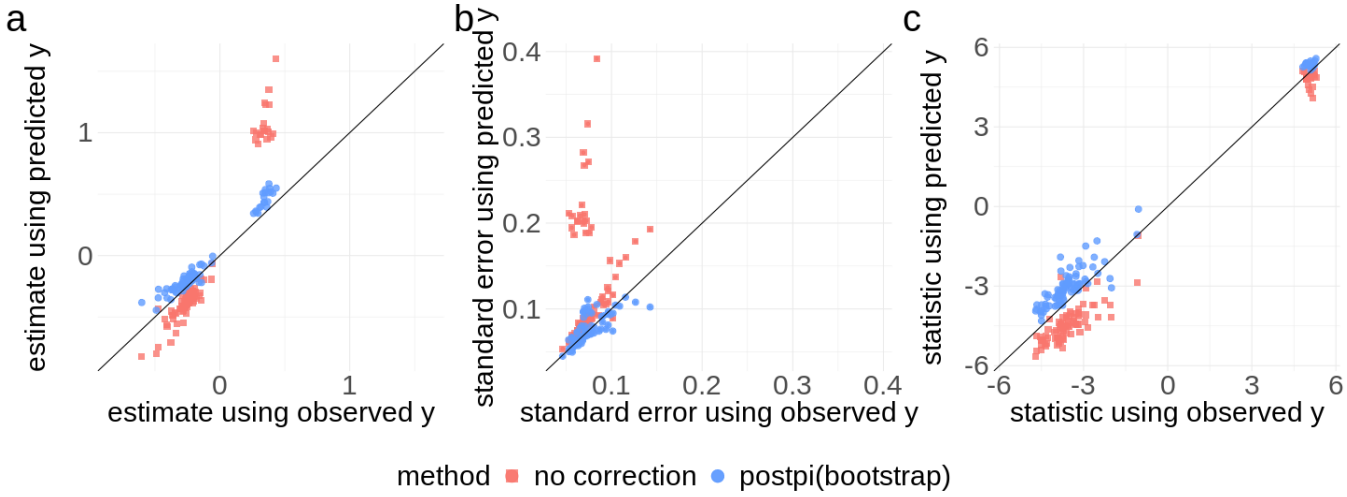

Figure 5: **Breast versus adipose tissue prediction.** Data were collected from the "Recount2" as described in Section 3.2.1. On the x-axis are the values calculated using the observed outcome y_i (i.e. $y_i = 1$ represents that breast tissue is observed) and on the y-axis are the values calculated using no correction (red color) and postpi bootstrap (blue color). We show (a) the estimates, (b) the standard errors and (c) the t-statistics. The postpi bootstrap approach improves the estimates, standard errors and t-statistics compared to no correction.

variables, and fit the inference model described above to each of the region on the validation set. We then get 101 estimates, standard errors and t-statistics. We compare them to the no correction approach as we did with the simulated data.

By comparing RMSE, we observed that the estimates, standard errors and test statistics are improved from no correction to postpi bootstrap. In Figure 5(a), RMSE of no correction (red color) is 0.36 compared to the truth and reduces to 0.08 with postpi bootstrap (blue color). The standard errors in Figure 5(b) have RMSE 0.076 for no correction (red color), but corrected to 0.01 for postpi bootstrap (blue color). The resulting t-statistics are improved from 0.91 for no correction (red color) to 0.63 for postpi bootstrap (blue color).

We also applied our approach to correct inference for models using predicted RNA-quality as an example of how to do post prediction inference for continuous outcomes (See Supplemental Section 2.1).

3.2.2 Describing cause of death distributions

We now move to our second example where the outcome of interest is the (predicted) cause of death and inputs are symptoms or circumstances reported by a caregiver or relative. Symptoms might include, for example, whether a person had a fever before they died, how long a cough lasted (if one was reported), or the number of times they visited a medical professional. We use data from the Population Health Metrics Research Consortium (PHMRC), which consists of about 7,800 "gold standard" deaths from six regions around the world. These data are rare because they contain both a physical autopsy (including pathology and diagnostic testing) and a verbal autopsy survey. Typically, only a small fraction of deaths will have an assigned cause (e.g. by a clinician reading the verbal autopsy survey) and these few labeled deaths will be used as inputs to train a model for the remaining deaths.

We split the data into a training and testing set, with 75% of the data used for training. The PHMRC data classify cause of death at several levels of granularity. For our experiments, we combined causes into twelve broad causes of death (Cancers, Diabetes, Renal diseases, Liver diseases, Cardiovascular causes, Stroke, Pneumonia, HIV/AIDS or Tuberculosis, Maternal causes, External causes, Other communicable diseases, and Other non-communicable diseases). We predicted the cause of death using *InSilicoVA*[11] which uses a Naive Bayes classifier embedded in a Bayesian framework to incorporate uncertainty between cause classifications.

In this example, we want to understand trends in the twelve combined causes of death and we care about both continuous and categorical symptoms. Continuous symptoms include age, number of people living at this address, age of the respondent. Categorical symptoms include year of death (2007, 2008, 2009, 2010), sex (male or female), death certificate issued (yes or no), used tobacco (yes or no), used alcohol (yes or no), education of deceased (College or Higher, High School, Primary School, No Schooling), separate room for cooking (yes or no), sex of respondent (male or female), education of respondent (College or Higher, High School, Primary School, No Schooling), region (AP, Bohol, Dar, Mexico, UP). The inference model we are interested in is: $g(E[y_i|SYM_i^j]) = \beta_0^j + \beta_1^j SYM_j$. Here $g(\cdot)$ is the logit link function for $j = 1, \ldots, 13$ (symptoms) and $i = 1, \ldots, n$, n is the total number of samples in the dataset. In this model, y_i represents one of the twelve combined causes at the ith sample and SYM_i^j is the jth symptom of interest on the ith sample.

For this dataset, we use categorical outcomes as the causes of death for the 1960 samples and

assume the outcomes are unobserved, as they typically would be in practice, for the remaining cases. Since the predicted values were obtained in a previously trained set using *InSilicoVA*[11], we only separate our data into a testing and validation set, each with a sample size $n = 980$. On the testing set, we fit a k-nearest neighbors model [30] to estimate the relationship between the observed cause of death and the probability of assigning the cause. On the validation set, we follow the ***Bootstrap Procedure*** in Section 2.3. Particularly in Step 3(ii), we simulate values from a distribution $\tilde{y}_i^b | pr_i^b \sim F_{\hat{\gamma}}$. In this example, we set F_γ to be a multinomial distribution with the probability parameters (i.e. probability of assigning each of the twelve broad causes of death) estimated from the relationship model as we did in the simulated data.

Among all the symptoms used to make causes of death prediction [11], we care about a subset of symptoms that also have balanced classes across the twelve broad causes of death to avoid highly variable model fits due to zero variance covariates. We then filter 13 symptoms we are interested in as regression variables and fit a logistic regression inference model to each of the selected symptom on the validation set. Because we include categorical regression variables with multiple factor levels in the inference model and get a inference result for each factor level, we obtain more factor level results than the number of symptoms. In total, we get 22 estimates, standard errors and t-statistics on the validation set. We then compare them to the no correction approach as we did with the simulated data.

By comparing RMSE, we observed that the estimates, standard errors and t-statistics are improved from no correction to postpi bootstrap (see RMSE table in Figure 6(d)). In Figure 6(a), RMSE of no correction (red color) is 0.84 compared to the truth and reduces to 0.42 with postpi bootstrap (blue color). The standard errors in Figure 6(b) have rmse 0.2 for no correction (red color), but corrected to 0.08 for postpi bootstrap (blue color). The resulting t-statistics are improved from 1.66 for no correction to 1.24 for postpi bootstrap (blue color).

4 Discussion

As machine learning becomes more common across a range of applications, it will become more common for predicted outcomes to be used as outcome variables in the subsequent statistical analyses. As we have shown, post-prediction inference can lead to highly variable or biased estimates of parameters of interest.

Here we introduced methods to correct for post-prediction inference and adjust point and interval

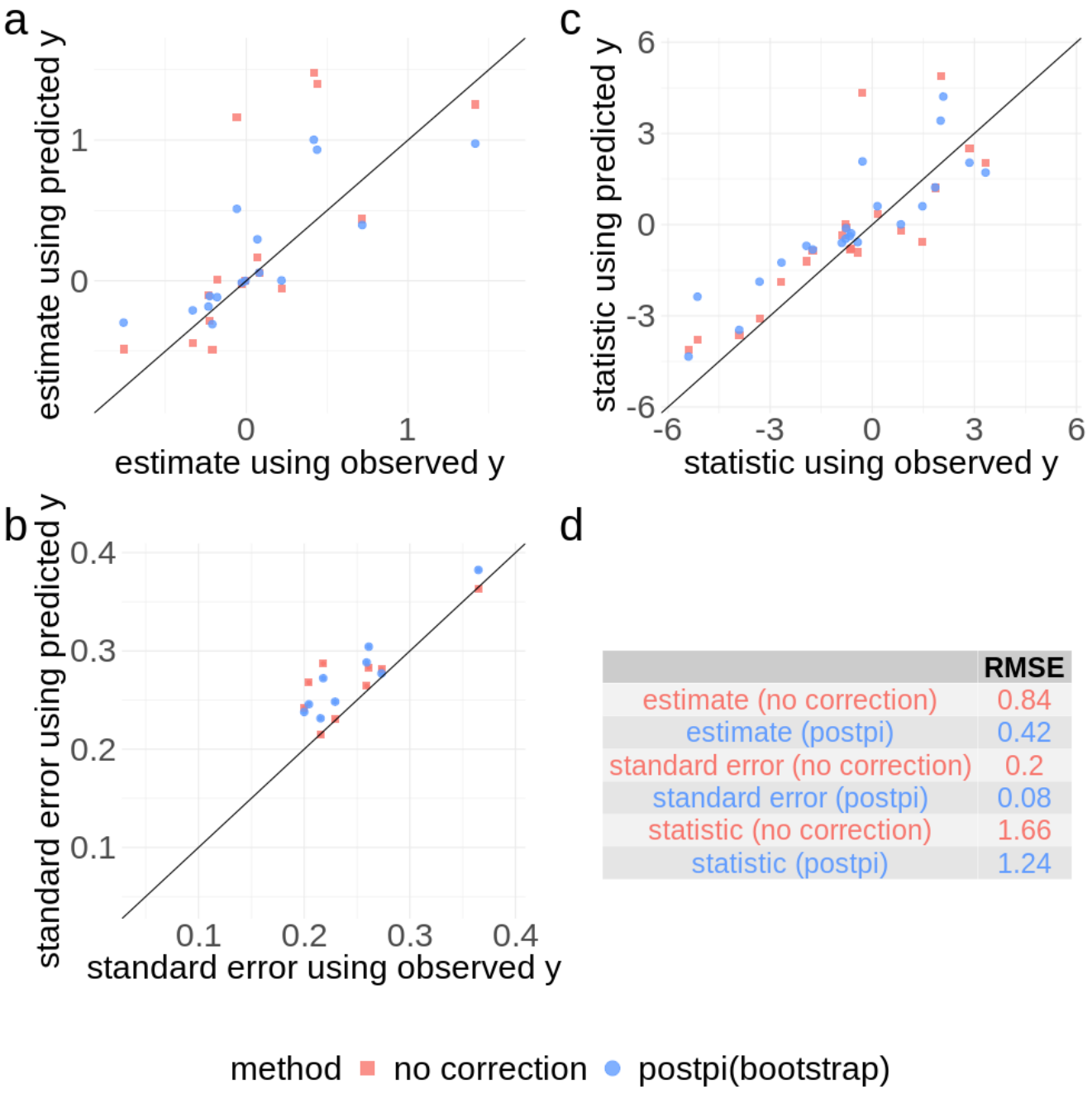

Figure 6: **Twelve causes of death prediction.** Data were collected from Population Health Metrics Research Consortium (PHMRC) described in Section 3.2.2. On the x-axis of the panel (a), (b), and (c) are the values calculated using the observed causes of death as outcome and on the y-axis are the values calculated using no correction (red color) and postpi bootstrap (blue color). We show (a) the estimates, (b) the standard errors, (c) the t-statistics and (d) the RMSE table. The postpi bootstrap approach improves the estimates, standard errors and t-statistics compared to no correction.

estimates when using predicted outcomes in place of observed outcomes. Our method is flexible enough to be applied to continuous and categorical outcome data, observed in fields such as medicine, public health, and sociology. Through simulated and real data, we show that our results outperform the most common current approach of ignoring the prediction step and performing inference without correction. By appropriately modeling the variability and bias due to the prediction step, the estimates, standard errors and test statistics are corrected towards the gold standard analysis we would obtain if we used the true outcomes.

Our approach relies on the key observation that the relationship between the observed and predicted values can be described as a simple model. While this observation is empirically true for the models and algorithms we considered, it may not hold universally. One limitation of our approach is that it depends on the fitness of the relationship model. For instance, when the predicted values are obtained from weak learners, the correlation between the observed and predicted outcomes is not strong, which may not be well captured by a simple model. Another limitation is that we assume the training, testing and validation sets follow approximately the same data generating distribution. If this assumption does not hold, inference performed on the bootstrapped values on the validation set will no longer reflect the true underlying data generating process. A potential solution is that we should first conduct data normalization using methods such as SVA [45], RUV [46] and removeBatchEffect in limma [47] to correct for latent confounders in the testing or validation sets. The normalized samples can then be input into our method for subsequent inferential analyses.

Despite these limitations, we believe correction for *post-prediction inference* is crucial for obtaining accurate inference when using outcomes produced by machine learning methods. Our correction represents the first step toward a general solution to the post-prediction inference problem. To make this method usable by the community we have released the *postpi* R package: [`https://github.com/SiruoWang/postpi`].

5 Acknowledgements

Research reported in this publication was supported by the National Institute of General Medical Sciences of the National Institutes of Health under award number R01GM121459.

References

[1] Jyotishman Pathak, Abel N Kho, and Joshua C Denny. Electronic health records-driven phenotyping: challenges, recent advances, and perspectives, 2013.

[2] Shannon E Ellis, Leonardo Collado-Torres, Andrew Jaffe, and Jeffrey T Leek. Improving the value of public rna-seq expression data by phenotype prediction. *Nucleic acids research*, 46(9):e54–e54, 2018.

[3] Samuel J Clark, Tyler McCormick, Zehang Li, and Jon Wakefield. Insilicova: a method to automate cause of death assignment for verbal autopsy. *arXiv preprint arXiv:1504.02129*, 2015.

[4] Rahul C Deo. Machine learning in medicine. *Circulation*, 132(20):1920–1930, 2015.

[5] Hamid Behravan, Jaana M Hartikainen, Maria Tengström, Katri Pylkäs, Robert Winqvist, Veli-Matti Kosma, and Arto Mannermaa. Machine learning identifies interacting genetic variants contributing to breast cancer risk: A case study in finnish cases and controls. *Scientific reports*, 8(1):13149, 2018.

[6] Jimmy Z Liu, Yaniv Erlich, and Joseph K Pickrell. Case–control association mapping by proxy using family history of disease. *Nature genetics*, 49(3):325, 2017.

[7] Alexander Gusev, Kate Lawrenson, Xianzhi Lin, Paulo C Lyra, Siddhartha Kar, Kevin C Vavra, Felipe Segato, Marcos AS Fonseca, Janet M Lee, Tanya Pejovic, et al. A transcriptome-wide association study of high-grade serous epithelial ovarian cancer identifies new susceptibility genes and splice variants. *Nature genetics*, 51(5):815, 2019.

[8] Manimozhiyan Arumugam, Jeroen Raes, Eric Pelletier, Denis Le Paslier, Takuji Yamada, Daniel R Mende, Gabriel R Fernandes, Julien Tap, Thomas Bruls, Jean-Michel Batto, et al. Enterotypes of the human gut microbiome. *nature*, 473(7346):174, 2011.

[9] Eric R Gamazon, Heather E Wheeler, Kaanan P Shah, Sahar V Mozaffari, Keston Aquino-Michaels, Robert J Carroll, Anne E Eyler, Joshua C Denny, Dan L Nicolae, Nancy J Cox, et al. A gene-based association method for mapping traits using reference transcriptome data. *Nature genetics*, 47(9):1091, 2015.

[10] Leonardo Collado-Torres, Abhinav Nellore, Alyssa C Frazee, Christopher Wilks, Michael I Love, Ben Langmead, Rafael A Irizarry, Jeffrey T Leek, and Andrew E Jaffe. Flexible expressed region analysis for rna-seq with derfinder. *Nucleic acids research*, 45(2):e9–e9, 2016.

[11] Tyler H McCormick, Zehang R Li, Clara Calvert, Amelia C Crampin, Kathleen Kahn, and Samuel J Clark. Probabilistic cause-of-death assignment using verbal autopsies. *Journal of the American Statistical Association*, 111(515):1036–1049, 2016.

[12] Muin J Khoury, Michael F Iademarco, and William T Riley. Precision public health for the era of precision medicine. *American journal of preventive medicine*, 50(3):398, 2016.

[13] Euan A Ashley. The precision medicine initiative: a new national effort. *Jama*, 313(21):2119–2120, 2015.

[14] Cathie Sudlow, John Gallacher, Naomi Allen, Valerie Beral, Paul Burton, John Danesh, Paul Downey, Paul Elliott, Jane Green, Martin Landray, et al. Uk biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS medicine*, 12(3):e1001779, 2015.

[15] Max Kuhn et al. Building predictive models in r using the caret package. *Journal of statistical software*, 28(5):1–26, 2008.

[16] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[17] Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M Dai, Nissan Hajaj, Michaela Hardt, Peter J Liu, Xiaobing Liu, Jake Marcus, Mimi Sun, et al. Scalable and accurate deep learning with electronic health records. *NPJ Digital Medicine*, 1(1):18, 2018.

[18] SA Khoury, D Massad, and T Fardous. Mortality and causes of death in jordan 1995-96: assessment by verbal autopsy. *Bulletin of the World Health Organization*, 77(8):641, 1999.

[19] Bin Yu and Karl Kumbier. Three principles of data science: predictability, computability, and stability (pcs). *arXiv preprint arXiv:1901.08152*, 2019.

[20] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[21] Donald B Rubin. *Multiple imputation for nonresponse in surveys*, volume 81. John Wiley & Sons, 2004.

[22] James Durbin. Errors in variables. *Revue de l'institut International de Statistique*, pages 23–32, 1954.

[23] Wayne A Fuller. *Measurement error models*, volume 305. John Wiley & Sons, 2009.

[24] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[25] Timothy A. Thomas, Ott Toomet, Ian Kennedy, Alex Ramiller, and University of Washington. The state of evictions: Results from the university of washington evictions project.

[26] Michel Garenne and Vincent Fauveau. Potential and limits of verbal autopsies. *Bulletin of the World Health Organization*, 84:164 – 164, 03 2006.

[27] JC Leitao, D Chandramohan, P Byass, R Jakob, K Bundhamcharoen, and C Choprapowan. Revising the WHO verbal autopsy instrument to facilitate routine cause-of-death monitoring. *Global Health Action*, 6(21518), 2013.

[28] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[29] Iñigo Barandiaran. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8), 1998.

[30] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[31] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[32] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[33] Llew Mason, Jonathan Baxter, Peter L Bartlett, and Marcus R Frean. Boosting algorithms as gradient descent. In *Advances in neural information processing systems*, pages 512–518, 2000.

[34] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018.

[35] Ben Hamner and Michael Frasco. *Metrics: Evaluation Metrics for Machine Learning*, 2018. R package version 0.1.4.

[36] Leonardo Collado-Torres, Abhinav Nellore, Kai Kammers, Shannon E Ellis, Margaret A Taub, Kasper D Hansen, Andrew E Jaffe, Ben Langmead, and Jeffrey T Leek. Reproducible rna-seq analysis using recount2. *Nature biotechnology*, 35(4):319, 2017.

[37] Abhinav Nellore, Leonardo Collado-Torres, Andrew E Jaffe, José Alquicira-Hernández, Christopher Wilks, Jacob Pritt, James Morton, Jeffrey T Leek, and Ben Langmead. Rail-rna: scalable analysis of rna-seq splicing and coverage. *Bioinformatics*, 33(24):4033–4040, 2016.

[38] Rasko Leinonen, Hideaki Sugawara, Martin Shumway, and International Nucleotide Sequence Database Collaboration. The sequence read archive. *Nucleic acids research*, 39(suppl_1):D19–D21, 2010.

[39] GTEx Consortium et al. Genetic effects on gene expression across human tissues. *Nature*, 550(7675):204, 2017.

[40] Frank W Albert and Leonid Kruglyak. The role of regulatory variation in complex traits and disease. *Nature Reviews Genetics*, 16(4):197, 2015.

[41] Harm-Jan Westra, Marjolein J Peters, Tõnu Esko, Hanieh Yaghootkar, Claudia Schurmann, Johannes Kettunen, Mark W Christiansen, Benjamin P Fairfax, Katharina Schramm, Joseph E Powell, et al. Systematic identification of trans eqtls as putative drivers of known disease associations. *Nature genetics*, 45(10):1238, 2013.

[42] Alexis Battle, Sara Mostafavi, Xiaowei Zhu, James B Potash, Myrna M Weissman, Courtney McCormick, Christian D Haudenschild, Kenneth B Beckman, Jianxin Shi, Rui Mei, et al. Characterizing the genetic basis of transcriptome diversity through rna-sequencing of 922 individuals. *Genome research*, 24(1):14–24, 2014.

[43] Matthew N McCall, Karan Uppal, Harris A Jaffee, Michael J Zilliox, and Rafael A Irizarry. The gene expression barcode: leveraging public data repositories to begin cataloging the human and murine transcriptomes. *Nucleic acids research*, 39(suppl_1):D1011–D1015, 2010.

[44] John Lonsdale, Jeffrey Thomas, Mike Salvatore, Rebecca Phillips, Edmund Lo, Saboor Shad, Richard Hasz, Gary Walters, Fernando Garcia, Nancy Young, et al. The genotype-tissue expression (gtex) project. *Nature genetics*, 45(6):580, 2013.

[45] Jeffrey T Leek and John D Storey. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS genetics*, 3(9):e161, 2007.

[46] Davide Risso, John Ngai, Terence P Speed, and Sandrine Dudoit. Normalization of rna-seq data using factor analysis of control genes or samples. *Nature biotechnology*, 32(9):896, 2014.

[47] Gordon K Smyth. Limma: linear models for microarray data. In *Bioinformatics and computational biology solutions using R and Bioconductor*, pages 397–420. Springer, 2005.