A Simulation Study

Patrick Lam

September 9, 2013

To test the ability of my model and the various matching procedures to recover individual causal effects and other quantities of interest, I conduct a simulation study to compare multiple methods. The simulation study generates toy data with the individual causal effects known and I evaluate the ability of the various matching methods to recover the ICEs on several evaluation criteria. I consider both continuous and binary dependent variables and evaluate the performance of the different matching methods as well as the different choices researchers must make with regard to the number of matches and the number of conditioning variables to include. The simulations suggest that in general, predictive mean matching seems to outperform other matching methods in recovering the ICEs.

1 Methods to be Compared

Recall the MCMC algorithm for the posterior of τ_i from before restated below. The simulations test various choices of $\mathcal{M}$ in step 1 of the algorithm. The choice of $\mathcal{M}$ consists of choosing a matching method, the number of matches used, and the set of variables to match on. To test the performance of different specifications of $\mathcal{M}$, I hold $\mathcal{M}$ constant each time, with the exception of possibly a random choice of the number of matches to use. Thus, in the simulation study, step 1 of the sampler is the same for each iteration within a single specification with the exception of specifications with random number of matches M. In those specifications, the number of matches varies across iterations but stays constant across i within the same iteration.

I define matching method to be the specification of the distance metric used and the method of picking matches given the distance metric. The four matching methods I consider are

1. **Mahalanobis matching**: The first distance metric I consider is the (squared) Mahalanobis dis-

MCMC Algorithm for the Posterior of τ_i

Repeat the following n_{sim} times:

Gibbs Sampler:
1. Draw a matching procedure $\tilde{\mathcal{M}}$ from $p(\mathcal{M})$.
2. Draw $\tilde{\theta}_{\mathcal{M}}$ from $p(\theta_{\mathcal{M}}|Y, X, W, D, \theta^{mis}, \mathcal{M})$.

for (i in 1:N){
3. Determine $\tilde{D}^{(i)}$ from matching procedure. **(matching step)**
4. Draw $\tilde{\theta}_i^{mis}$ to estimate θ_i^{mis}.
}

Draw from PPD and Calculate τ_i:
for (i in 1:N){
5. Draw $\tilde{Y}_i^{mis}$ from $f(\cdot|\tilde{\theta}_i^{mis})$. **(imputation step)**
6. Calculate $\tilde{\tau}_i = W_i(Y_i - \tilde{Y}_i^{mis}) + (1 - W_i)(\tilde{Y}_i^{mis} - Y_i)$.
}

tance metric used in Rubin (1980). The Mahalanobis distance between two observations with covariate values X_1 and X_2 is

$$\Delta_M(x_1, x_2) = \sqrt{(X_1 - X_2)^T S^{-1}(X_1 - X_2)}$$

where S^{-1} is the sample covariance matrix of X. For τ_i, I calculate the squared Mahalanobis distance between X_i and X_j, $\forall W_i \neq W_j$ and then use the M nearest neighbors as matches. Unlike the remaining matching methods, Mahalanobis matching is model-free in the sense that it only looks at the in-sample covariate distances rather than imposing a parametric model.

2. **predictive mean matching**: Since the goal of estimating ICEs is to impute the missing potential outcomes with matching, one way to do this is to first model the means of the observed outcomes in the treatment and control groups and match based on the model. Let i index any treated observation. Then the best imputation of $Y_i(0)$ is likely to come from control observations with observed outcomes that are closest to $Y_i(0)$. Denote Y_c and X_c as the observed outcomes and covariates for the control group and Y_t and X_t as the analogous for the treatment group. Since $Y_i(0)$ is unobserved, I first make a best guess of $Y_i(0)$ by modeling the outcomes for the control group with a linear regression of Y_c on X_c.[1] Let θ_c denote the vector of parameters (β_c, σ_c^2) from this regression. I then calculate a predictive mean score for all observations as

$$\tilde{\mu}_{(c)} = X\tilde{\beta}_c$$

[1]For now, I assume that the covariates enter the regression linearly without any interactions or polynomials.

Note that $\tilde{\mu}_{(c)}$ is calculated for all observations and the subscript refers only to the fact that the predictive mean score is calculated from $\tilde{\beta}_c$. For treated observation i, use the M nearest neighbor control observations on $\tilde{\mu}_{(c)}$ as its matches.[2] $\tilde{\mu}_{i,(c)}$ basically serves as our best initial guess of $Y_i(0)$ based on a regression model.

Now let j index any control observation. To estimate τ_j, I do predictive mean matching with a similar process. Regress Y_t on X_t to get an estimate of θ_t, which consists of (β_t, σ_t^2). Calculate another predictive mean score for all observations as

$$\tilde{\mu}_{(t)} = X\tilde{\beta}_t$$

For control observation j, use the M nearest neighbor treated observations on $\tilde{\mu}_{(t)}$ as its matches. $\tilde{\mu}_{i,(t)}$ serves as the initial guess of the missing $Y_j(1)$. In essence, one can think of this process as conducting predictive mean matching twice with the treatment indicators reversed the second time.

Within the MCMC algorithm, predictive mean matching involves drawing $\theta_{\mathcal{M}} = \{\beta_t, \beta_c, \sigma_t^2, \sigma_c^2\}$ in the second step with a Gaussian linear regression. For priors, I use

$$\begin{aligned} \beta_w &\sim \text{improper uniform} \\ \sigma_w^2 &\sim \mathcal{IG}\left(\frac{0.001}{2}, \frac{0.001}{2}\right) \end{aligned}$$

for $w = t, c$. Since these parameters only depend on $\mathcal{M}$ and the observed data, the full conditionals to draw from are simply the conditional distributions in a Gaussian linear regression.

$$\begin{aligned} \beta_w|\sigma_w^2, Y_w, X_w, \mathcal{M} &\sim \mathcal{N}(m^*, V^*) \\ V^* &= \left(X_w'(\sigma_w^2 I)^{-1}X\right)^{-1} \\ m^* &= V^*(X_w'(\sigma_w^2 I)^{-1}Y_w) \end{aligned}$$

$$\begin{aligned} \sigma_w^2|\beta_w, Y_w, X_w, \mathcal{M} &\sim \mathcal{IG}\left(\frac{\nu^*}{2}, \frac{\delta^*}{2}\right) \\ \nu^* &= n_w + 0.001 \\ \delta^* &= (Y_w - X_w\beta_w)'(Y_w - X_w\beta_w) + 0.001 \end{aligned}$$

for $w = t, c$ where n_w is the number of observations in treatment group w. Step 3 of the algorithm

[2] One can also match $\tilde{\mu}_{(c)}$ with the actual observed control outcomes although it will be more difficult to differentiate between good matches with discrete outcome variables.

uses the draw of $\theta_{\mathcal{M}}$ at each iteration to find matches for each observation through the predictive mean matching process described. The benefit of predictive mean matching is that the distance measure is most directly related to the quantity of interest of the missing potential outcomes. With a large enough sample, predictive mean matching should produce balance between an observation and its matches since observations with the same observed covariate values should have the same predictive mean up to some degree of randomness. Predictive mean matching reverses the process by assuming that observations with similar predictive means should have similar observed covariate values.

3. **propensity score matching**: The propensity score is defined as the conditional probability of being assigned to treatment given a vector of covariates X. Under randomized treatment assignment, the propensity score should be a known function whereas in observational studies, the propensity score is unknown and must be estimated. The propensity score reduces the dimensions of X down to a scalar and Rosenbaum and Rubin (1985) show that adjusting for the propensity score is sufficient for producing unbiased estimates of treatment effects. Furthermore, they show that adjusting for the sample estimate of the propensity score can produce balance on the covariates in the sample.

   Define the propensity score for observation i as

$$e_i = P(W_i = 1|X_i)$$

   I estimate the propensity scores for all observations using a logistic regression within the MCMC algorithm. In step 2 of the algorithm, let $\theta_{\mathcal{M}}$ be the coefficients β from a Bayesian logistic regression of W on X.[3] Our estimated propensity scores take the form

$$\tilde{e}_i = \frac{1}{1 + \exp(-X_i\tilde{\beta})}$$

   Note that the propensity scores are a function of draws from the posterior of the regression. For each draw of $\tilde{\beta}$, I calculate a propensity score $\tilde{e}_i(X_i)$ for each individual. Since the propensity scores are unknown and estimated, this incorporates uncertainty over the propensity scores, an approach similar to that in An (2010). For matching, I use nearest neighbor matching on the linear propensity score

$$\ln\left(\frac{\tilde{e}_i}{1 - \tilde{e}_i}\right) = X_i\tilde{\beta}$$

[3] Again, for now X enters into the propensity score equation linearly.

which has been found effective for reducing bias in the matching literature (Rubin, 2001). For each observation i, matches are produced by taking the M observations in the opposite treatment group with the closest linear propensity score. Observations may be used as donors to multiple other observations, but can only be used once for any particular observation.

Within the MCMC algorithm, estimating a logistic regression in step 2 requires embedding a Metropolis-Hastings step. I use an improper uniform prior on β and a random walk Metropolis algorithm.

4. **subclassification** (on the linear propensity score): In addition to nearest neighbor matching on the linear propensity score, I also consider subclassification on the linear propensity score. The idea behind subclassification is to sort the estimated propensity score and then divide the observations into M subclasses based on the ordered propensity scores.[4] Rosenbaum and Rubin (1984) show that subclassification on the propensity score with as few as five subclasses can substantially reduce bias in estimating treatment effects. Much like choosing the number of matches, choosing the number of subclasses is part of the choice of $\mathcal{M}$ in the algorithm. I consider both fixed and random M in my simulations. Within the algorithm, the linear propensity scores are estimated exactly as above, and the subclassification affects the choice of which observations contribute to the donor pool $\tilde{D}_i$. Observations in the same subclass as the observation to be matched are considered to be a part of the donor pool. I restrict the analyses to contain at least two treated and two control observations in every subclass. Because the linear propensity scores are estimated stochastically, within any specific iteration, it is possible to have subclasses that do not contain at least two treated and two control observations. In those rare instances, I decrease M by one for that iteration of the algorithm only until every subclass in that iteration meets the criteria.

The simulations presented compare the choice of one of these methods as well as the number of matches/subclasses and the set of variables to match on. All of these choices are captured in $\mathcal{M}$ in step 1 of the algorithm. As mentioned before, each simulation holds constant the choice of method and number of variables to match on. The number of matches/subclasses are either held constant or allowed to vary randomly within a range. Within a single iteration in a simulation, steps 1 and 2 produce a donor pool for every observation i, which is denoted $\tilde{D}_i$ in step 3. Using the donor pool, I then draw a value of $\tilde{\theta}_i^{mis}$ in step 4 by modeling the mean of the donor pool. For continuous outcome variables, I draw $\tilde{Y}_i^{mis}$ from the posterior predictive distribution $\mathcal{N}(\theta_i^{mis}, \sigma_i^{2\,mis})$.[5] For binary outcome variables, I

[4] In the context of subclassification, I use M to refer to the number of subclasses rather than the number of matches. Increasing M actually decreases the number of subclasses holding sample size constant.

[5] $\sigma_i^{2\,mis}$ is also estimated from the donor pool with an $\mathcal{IG}\left(\frac{0.001}{2}, \frac{0.001}{2}\right)$ prior.

draw $\tilde{Y}_i^{mis}$ from a Bern(θ_i^{mis}) distribution.

In addition to the four matching methods, I also consider two methods which do not use a matching procedure as a baseline.

1. (Bayesian) **regression imputation**: I take the simplest and most commonly used case where the imputations of the missing potential outcomes are generated from the coefficients of a Bayesian linear regression model. I fit a regression model of Y on W and X using the priors

$$\begin{aligned} \beta &\sim \text{improper uniform} \\ \sigma^2 &\sim \mathcal{IG}\left(\frac{0.001}{2}, \frac{0.001}{2}\right) \end{aligned}$$

   The missing potential outcomes are then imputed from the coefficients $\tilde{\beta}$ such that

$$\tilde{Y}_i^{mis} = \tilde{\beta}_0 + \tilde{\beta}_1(1 - W_i) + \tilde{\beta}_X X_i$$

   where β_0 is the intercept, β_1 is the coefficient on W, and β_X is the set of coefficients on X from the regression. Since I use fairly uninformative priors, the estimates from this Bayesian regression will be nearly identical to estimates from a non-Bayesian regression. I use a Bayesian regression simply to remain consistent with the other approaches I test. I use this model as a baseline since this is probably the simplest and most common regression model-based way to impute potential outcomes. Note that the imputations here come solely from an estimate of an average treatment effect.

2. **no matching (all)**: I consider the case where all of the j observations where $W_i \neq W_j$ are used as matches for i. In this specification, no matching algorithm is used since all observations of the other treatment group are used as matches. In the case where treatment assignment is randomized, one would expect that no matching would produce roughly the same quality of matches as other matching algorithms. The donor pool for this method is simply all observations with a different treatment status and the estimation of θ_i^{mis} and imputation of Y_i^{mis} follows the same process as the matching procedures above.

Within each method, I also test the sensitivity of the choice of the number of matches to use and the set of covariates to include where appropriate. Thus, for each specification of $\mathcal{M}$ that I test, I vary all

three dimensions that the researcher can choose.

2 Setting Up the Simulations

In general, I will only discuss how I perform the simulations and the results for the case of a continuous dependent variable. I also repeat some of the simulations for a binary dependent variable, but the results are similar so I relegate those simulations to the appendix.

Data generating processes

To assess the performance of the different methods, I generate fake data from numerous linear and non-linear data generating processes to test how well the methods recover various causal estimands of interest. The data generating processes are borrowed from the ones used by Hainmueller (2012) and Frölich (2007) with a few changes tailored specifically to the framework used here. The best performing method(s) should ideally be fairly robust to deviations from non-linearity in the data generating process even though I only use linear specifications. I also consider three different sample sizes of 100 (small), 1000 (medium), and 5000 (large).

To begin, I generate ten covariates that completely determine the outcomes:

- $x_1 \sim \mathcal{N}(0, 2^2)$
- $x_2 \sim \mathcal{N}(0, 1)$
- $x_3 \sim \mathcal{N}(0, 1)$
- $x_4 \sim \mathcal{U}(-3, 3)$
- $x_5 \sim \chi^2_1$
- $x_6 \sim Bernoulli(.5)$
- $x_7 \sim \mathcal{N}(0, 1)$
- $x_8 \sim \mathcal{N}(0, 1)$
- $x_9 \sim \mathcal{N}(0, 1)$
- $x_{10} \sim \mathcal{N}(0, 1)$

Using these ten covariates, I generate the potential outcome $Y_i(0)$, the outcome without treatment, for each observation i. I consider three different outcome generating equations:

1. $Y(0) = x_1 + x_2 + x_3 - x_4 + x_5 + x_6 + x_7 - x_8 + x_9 - x_{10}$
2. $Y(0) = x_1 + x_2 + 0.2x_3x_4 - \sqrt{x_5} + x_7 + x_8 - x_9 + x_{10}$
3. $Y(0) = (x_1 + x_2 + x_5)^2 + x_7 - x_8 + x_9 - x_{10}$

The three equations vary in their degree of linearity, starting from a (1) linear relationship between Y and X and going to (2) a moderately non-linear and (3) very non-linear relationship. For each i, I then assign treatment in three different ways:

1. $p(W = 1) = 0.5$
2. $\eta = x_1 + 2x_2 - 2x_3 - x_4 - 0.5x_5 + x_6 + x_7$
   $W = 1$ if $\eta > 0$; otherwise $W = 0$
3. $\eta = 0.5x_1 + 2x_1x_2 + x_3^2 - x_4 - 0.5\sqrt{x5} - x_5x_6 + x_7$
   $W = 1$ if $\eta > 0$; otherwise $W = 0$

In the first case, treatment assignment is completely random with equal probability of being assigned treatment or control. In the second case, treatment assignment is linearly related to the first seven covariates. Since in my framework, there exists a set of covariates $X^{(p)}$ that completely explain the outcomes, I also allow a subset of the covariates (the first seven covariates) to be confounders that perfectly predict treatment assignment. In the third case, the first seven covariates are non-linearly related to treatment assignment. Note that in scenarios 2 and 3, conditioning on $x1$ through x_7 is sufficient to control for confounders. The three outcome equations and the three treatment assignment scenarios create nine different data combinations that range from unconfounded and linear in Y to (linear and non-linear) confounded treatment assignment and very non-linear in Y.

For most specifications, I draw each "true" τ_i independently from a $\mathcal{N}(2, \sqrt{3}^2)$ distribution. Drawing τ_i independently gives the most general situation in which each individual's τ_i gives no information about any other τ_i. If one considers the case where treatment effect heterogeneity is explained by some observed covariate, then matching on that covariate should improve the ability of the model to capture the different τ_i. Thus, drawing the true τ_i independently serves as a conservative test of the methods' ability to estimate the individual causal effects. In a few other specifications, I also vary the distribution from which τ_i is drawn. Specifically, I consider cases where the τ_i are drawn independently from:

1. $\mathcal{N}(2, \sqrt{3}^2)$
2. $\mathcal{N}(20, \sqrt{3}^2)$
3. $\mathcal{N}(2, \sqrt{100}^2)$
4. $\mathcal{N}(20, \sqrt{100}^2)$
5. mixture of $\mathcal{N}(2, \sqrt{3}^2)$ and $\mathcal{N}(20, \sqrt{3}^2)$ with equal probability on each
6. mixture of $\mathcal{N}(2, \sqrt{100}^2)$ and $\mathcal{N}(20, \sqrt{100}^2)$ with equal probability on each

By varying the mean of the τ_i distribution, I vary the size of the effects to see how well the methods perform as the effect sizes increase. I vary the standard deviation of the τ_i distribution to test how well the methods perform over a changing range of τ_i. I expect the methods to perform better with greater effect sizes (more power) and a smaller range over τ_i (less heterogeneity). I also consider the mixture distributions to simulate scenarios in which treatment effects are clustered such that treatment has a range of effects for one group and a different range of effects for another group. For example, treatment may hurt one group of individuals and help another group.

To complete the data generating processes, I generate $Y_i(1)$:

$$Y_i(1) = Y_i(0) + \tau_i$$

I then put together the "observed" dataset that the models use. To mirror the typical data analysis, I run the different model specifications that I test using the datasets containing the following variables:

- W
- $Y = W \times Y(1) + (1 - W) \times Y(0)$
- $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}$

Causal estimands of interest to be recovered

Since I generate the individual causal effects τ_i in the simulations, τ_i and any other causal estimand is known. The goal of the simulations is to evaluate how well a method can recover the known true

values of these estimands. I consider how well a method recovers the following causal estimands in the simulations:

- **Individual causal effects**: The most important estimands to recover are the N τ_i's themselves. For the case of binary dependent variables, τ_i can only take on values of -1, 0, and 1 so it is more difficult to actually evaluate how well the methods recover the τ_i since the posterior distribution is a mixture of two of the possible three values. Therefore, I only look at the aggregated estimands described below for the simulations with binary dependent variables.
- **Average treatment effect**: Another important quantity to recover is the ATE. Any method that can recover the ICEs should be able to recover the ATE correctly since the ATE is a simple linear function of the ICEs. Since the ATE is usually the easiest estimand to estimate, any method that performs poorly on recovering the ATE is probably not a very robust and useful method.
- **Average treatment effect on the treated**: The ATT is another average effect that calculates the average effect over a subset of the data. Since recovering the ICEs correctly implies recovering any aggregation of the ICEs, I should be able to randomly choose any subset and calculate the average effect and judge a method by its ability to recover this average effect.
- **Treatment effect quantiles** (0.5, 0.75, 0.95): Since I claim that estimating ICEs allows for unparalleled flexibility in recovering any other causal estimand, I put the method to a difficult test by attempting to recover the treatment effects at different quantiles. To calculate a quantile treatment effect, I sort the ICEs from lowest to highest and then take the desired quantile of these sorted effects. Even though my simulations have even numbered sample sizes, I take the quantiles without averaging, so the 0.75 quantile treatment effect for $N = 1000$ is the 750th ordered statistic for the sorted τ_i. As the quantiles become more extreme, I expect any method to perform worse so my model should recover the 0.5 quantile with more accuracy and precision than the 0.75 and 0.95 quantiles. The results available in the appendix confirm this to be true.

Performance metrics used to evaluate the methods

The typical simulation study uses performance metrics such as bias, mean squared error, confidence interval coverage, or power to evaluate a statistical method. All of these metrics stem from a frequentist perspective where the data is assumed to be sampled randomly many times and each time the method

calculates a statistic that characterizes the sampled data. All the metrics used are concerned with how the method performs on average over repeated samples. These traditional metrics are inappropriate in the current context for two reasons. First, the method I propose is fundamentally a Bayesian method that does not rely on a repeated sampling framework. Instead, the data is assumed to be sampled once and a Bayesian method conditions on the actual observed dataset only, so using traditional metric to test the repeated sampling properties of a Bayesian method makes little sense. Second, the whole idea of individual causal effects as I present them here is incompatible with a repeated sampling framework. My framework assumes that the potential outcomes are fixed. Therefore, the estimand does not change regardless of how many times you sample. τ_i remains the same for individual i even if i was sampled repeatedly. Furthermore, since individual causal effects are specific to individual i, a repeated sampling framework would involve sampling i such that i appears in the dataset for some samples and not others. For samples that do not include i, τ_i is unestimable. Therefore, I cannot use traditional notions of repeated sampling to evaluate the methods proposed.

Instead, I develop and use Bayesian versions of bias, mean squared error, power, and coverage. Under the Bayesian version, I replace the repeated sampling framework by evaluating the methods over the N individuals in the dataset. For example, instead of evaluating how a method performs on average over repeated samples, I evaluate how a method performs by averaging over the N individuals observed. The Bayesian metrics that I use for ICEs and other causal estimands of interest include posterior mean bias, expected error loss, the proportion of the credible intervals not including 0, and calibration coverage.[6]

- **Posterior mean bias** ("bias"): Let θ be any estimand or parameter of interest. The traditional bias of an estimator $\hat{\theta}$ is

$$\text{bias}(\hat{\theta}) = E(\hat{\theta}) - \theta$$

where the expectation is taken over $\hat{\theta}$ under repeated samples of the data. $\hat{\theta}$ is usually some "best" estimate of θ. Contrast this with the posterior mean bias metric that I use.

$$\text{posterior mean bias} \quad = \quad E(\theta|X) - \theta$$

where X represents the observed data and $\theta|X$ is the posterior distribution of θ conditional on the observed data. The expectation here is the expectation of the posterior distribution, or the

[6]For the simulations with binary continuous variables, I only look at posterior mean bias and expected error loss because the latter two metrics are difficult to calculate when τ_i only takes on discrete values of -1, 0, and 1.

posterior mean. Using decision theory and a quadratic loss function, it can be shown that the posterior mean is the Bayes estimator in that it minimizes the expected loss given θ.[7] Therefore, the posterior mean bias is a Bayesian analogue of bias in the frequentist sense. It represents a broad notion of how far off from the truth our "best" estimate is. For the aggregated estimands such as the ATE or ATT, posterior mean bias is calculated simply as the mean of the MCMC simulations from the posterior distribution minus the true value of the estimand calculated from the τ_i, which are generated from a known data generating process. For the ICEs themselves, I can look at the posterior mean bias for each of N τ_i's, but I choose to summarize them by the average[8] and standard deviation of the N posterior mean biases to make comparing the methods easier.

- **Expected error loss** ("root mse"): For any parameter θ and estimator $\hat{\theta}$, the typical root mean squared error calculation is

$$\sqrt{E[(\hat{\theta} - \theta)^2]} \quad = \quad \sqrt{\text{variance} + \text{bias}^2}$$

again with the expectation taken over repeated samples. The root mean squared error gives a rough estimate of how far off the estimator is from the truth, taking into account both bias and variance. The Bayesian analogue that I use is the expected error loss, which does not require expectations over repeated samples. For notational clarity, now let θ denote a random variable for the parameter of interest and let θ^* denote the true underlying value of the parameter.[9] Then

$$\text{expected error loss} \quad = \quad \sqrt{\int (\theta - \theta^*)^2 p(\theta|X) d\theta}$$

[7] In decision theory, one must take an action or make a decision a assuming that the true state of nature is θ. Using a quadratic loss function $L(\theta, a) = (\theta - a)^2$, the expected loss given our posterior is

$$\begin{aligned} E_{\theta|X}[L(\theta, a)] &= \int (\theta - a)^2 p(\theta|X) d\theta \\ &= \int \theta^2 p(\theta|X) d\theta - 2a \int \theta p(\theta|X) d\theta + a^2 \int p(\theta|X) d\theta \\ &= E(\theta^2|X) - 2aE(\theta|X) + a^2 \end{aligned}$$

One can minimize the loss by differentiating with respect to a and setting it equal to zero, giving us the posterior mean as the decision or estimate that minimizes expected loss.

$$\hat{a} = E(\theta|X)$$

[8] Averaging over the N posterior mean biases for the τ_i is actually equivalent to looking at the posterior mean bias for the ATE due to the linearity of expectations.

[9] The notation used in this section may be confusing because I attempt to compare frequentist and Bayesian methods assuming a fixed underlying true parameter, which is usually reserved only for frequentists. Bayesians usually describe parameters probabilistically using random variables even though a true underlying parameter value may exist. Since I am comparing estimates of θ from Bayesian models to a true value of θ in my simulations, I assume a fixed parameter value. To clarify the notation, whenever I discuss frequentist methods, θ is the fixed parameter value. When discussing Bayesian methods, θ can refer to the random variable for the parameter or the true underlying value given by nature. I attempt to be more explicit by using θ^* to represent the true underlying value when discussing both the random variable and the true underlying value.

In contrast to the posterior mean bias metric, the expected error loss metric accounts for the deviations from θ^* for all possible values of θ rather than just the point estimate at the posterior mean. It is basically a weighted average of the squared error loss for the entire support of the posterior. In practice, the expected error loss is calculated by taking each draw $\tilde{\theta}$ from the posterior and calculating its squared error relative to θ^* and then taking the average across the draws. For aggregate estimands like the ATE, I look at the expected error loss whereas for the N τ_i's, I look at the average of the N expected error losses.

- **Proportion of the credible intervals[10] not including 0** ("power"): In hypothesis testing, the typical definition of the power of a statistical method is the probability of the method rejecting the null hypothesis given that the null hypothesis is false. In other words, it is the probability of detecting an effect when one exists or the probability of not committing a Type 2 error. The statistical power of a method depends on the statistical significance criteria used (α level), the magnitude of the effect or the effect size, and the sample size. Using the typical $\alpha = 0.05$ criteria, one would usually test the statistical power with simulation by randomly drawing data with the same sample size and the same predefined effect size that matches the alternative hypothesis[11], calculating the statistic or test for each sample, and then determining the proportion of samples in which the test rejects the null hypothesis (e.g. the proportion of times that the test "gets it right"). One direct way is to calculate the proportion of 95% confidence intervals that do not contain the null hypothesis. This proportion is a calculation of the statistical power given the specified α, effect size, and sample size.

  For the application of my Bayesian model to the estimation of ICEs, I cannot use the typical way to calculate power because as described earlier, there is no repeated sampling principle on which to rely. Instead, I rely on the N observations in the simulated dataset with N ICEs as N "repeated samples." I then calculate the proportion of 95% credible intervals for the N τ_i's that do not include 0 as a rough estimate of the "power" for a particular method. The estimate is rough and does not exactly satisfy the definition of power in the typical sense. Assuming that the null hypothesis is $\tau_i = 0$,[12] the data generating process for the case of continuous outcome variables always generates $\tau_i \neq 0$ for all i, which satisfies the condition of the null hypothesis being

[10] I use 95% credible intervals here and throughout to refer to the central 95% region of the posterior to be consistent with the idea of a 95% confidence interval. The interpretation of a 95% credible interval is that the truth lies in the interval with probability 0.95. In practice, I calculate the 95% credible intervals by simply taking the 0.025 and 0.975 quantiles of the posterior draws.

[11] If the null hypothesis is that the effect size is zero and the alternative hypothesis is that the effect size is not equal to zero, then the effect size in the simulations is set to a value that is not equal to zero.

[12] The language here is not exactly correct since I am using hypothesis testing language in a Bayesian context. Nevertheless, I use this language of testing for power because I want to compare the performance of different methods in capturing the effects when they exist.

false.[13] However, unlike the case of the typical power calculation, τ_i is not constant for all i, so the proportion is calculated over varying effect sizes. Nevertheless, given my framework and goals, this calculation gives a rough estimate of power which will approach the more traditional power calculation as the standard deviation on the τ_i approaches 0.

- **Calibration coverage** ("coverage"): The way typical simulation studies assess the accuracy of confidence intervals generated by a method is by looking at its coverage probability, which is the proportion of the time that the interval contains or "covers" the true value of the parameter. Recall the correct definition of a confidence interval, say the (nominal) 95% confidence interval, is that in repeated samples, 95% of the calculated 95% confidence intervals should contain the truth. Ideally then, the actual coverage probability of the method equals the nominal probability of 0.95. Deviations from 0.95 would suggest that some assumptions of the model are not met. To derive the coverage probability in a simulation, one would simulate repeated samples from the data generating process, holding the parameter at a single "true" value, calculate the 95% confidence interval each time, and then calculate the proportion of the confidence intervals that contain the "true" value.

  Under my Bayesian framework for estimating ICEs, repeated sampling once again does not make sense because of the Bayesian and the ICE aspects. Much like the "power" calculation, I once again leverage the N τ_i's as a substitute for repeated sampling. Here I appeal to the idea of Bayesian calibration with credible intervals. A Bayesian 95% credible interval has a much more intuitive definition as the interval in which the true value occurs with 0.95 probability. Probability here is subjective since it is a function of both the data and the subjective prior probability. However, the idea of calibration is that the Bayesian model should produce a 95% credible interval that is calibrated such that it can predict 95% of future observations correctly. Applying this logic to the simulation for ICEs, a method that performs well should have 95% credible intervals that contain the true values 95% of the time. In my simulations, I calculate the proportion of the N 95% credible intervals that contain the true ICEs. Note that as in the calculation of the rough "power" statistic above, each τ_i varies, which differs from the traditional coverage calculations. However, with Bayesian calibration, each ICE 95% credible interval should ideally contain its own τ_i with 0.95 probability, so I can look across all N τ_i and estimate the proportion that contain its own true τ_i as the calibration coverage probability. The best performing methods are the ones that have coverage probability closest to 0.95 using the 95% credible intervals in the calculation.

[13] This is due to the fact that τ_i is continuous and the probability of drawing any specific value is 0 for a continuous distribution.

I assess the performance of the different matching methods and specifications using all four of these metrics when possible. Each metric conveys a different aspect of model performance and the methods that perform the best ideally perform well on all four metrics.

Different specifications

As alluded to above, I test the ability of the model and different matching methods in estimating the causal estimands of interest. For the first set of simulations, I run numerous simulations of the model, each time varying one aspect of the model specification or one aspect of the data generating process. A model specification includes

- **choice of method**: regression, all, mahalanobis, predictive mean, propensity score, or propensity score subclassification
- **number of matches** (for mahalanobis, predictive mean, and propensity score) **or number of subclasses** (for propensity score subclassification): small, medium, large, or random[14]
- **number of X variables to condition on**: 0 (for the method all only), 5, 7, or 10[15]

In addition to varying the model specifications, I also vary the data generating process for each specification. The data generating process specifications are

- **sample size**: 100, 1000, or 5000[16]
- **outcome generating equation**: linear, moderately non-linear, or very non-linear
- **treatment assignment**: unconfounded, confounded linearly, confounded non-linearly

[14]For the number of matches, small, medium, large, and random were defined as 2, 10, 25, and an integer uniformly drawn from the range 2 through 25 respectively. For the number of subclasses, small, medium, large, and random were defined differently depending on the sample size for each simulation. With sample size of 100, the number of subclasses used was 2,4,5, and an integer uniformly drawn from the range 2 through 5. With sample size of 1000, the number of subclasses used was 5,10,20, and an integer uniformly drawn from the range 5 through 20. With sample size of 1000, the number of subclasses used was 5,20,50, and an integer uniformly drawn from the range 5 through 50.

[15]The variables were conditioned on in order, so 5 X variables conditioned on means conditioning on x_1 through x_5 and so forth. Recall that for the confounded treatment assignments, the first 7 X variables were used in the confounding.

[16]When increasing the sample size, rather than regenerating a new dataset completely, I keep the previous sample and simply add on extra observations, so a dataset with sample size 1000 contains 100 observations from the previous simulation and adds 900 new observations. By adding on observations instead of regenerating completely new observations, I allow the datasets of different sizes to be comparable (conditional on the same generating equations) because the first 100 observations in the dataset are the same across the two sizes. I retain the condition that these "individuals" are the same, which is more coherent given the ICE framework.

For this first set of simulations, I hold the distribution of τ_i constant by generating them all from a $\mathcal{N}(2, \sqrt{3}^2)$ distribution. There are 52 combinations of model specifications and 27 combinations for the data generating processes, which lead to $52 \times 27 = 1404$ different simulations in the first set.

I then consider a second set of simulations that further tests the optimal number of matches to use. I hold the data generating sample size to 1000 with the nine different outcome/treatment assignment generating equations and only condition on 7 covariates. I only consider the case of predictive mean matching. The specification that varies is the number of matches, which I now specify as a percentage of the smaller treatment group. Given a simulated dataset, I take smaller of the treated or control groups and calculate the number of matches M as a percentage of this number (rounded up). The percentages I consider are

- every 1 percentage point between 1% and 9% inclusive
- every 10th percentage percentage point between 10% and 90% inclusive
- the case of 100%, which I then make equivalent to just the "all" matching method (so the 100% here is actually 100% of both treatment groups)

The different percentages produce 19 different specifications, combined with the 9 different data generating processes to produce $19 \times 9 = 171$ different simulations in the second set.

Finally I consider a third set of simulations to assess the sensitivity of the results to different ways of generating the true values of τ_i as I described above. I hold the sample size to 1000 again with the nine different outcome/treatment assignment generating equations, condition on only 7 covariates, and restrict the number of matches or subclasses to 25 (except for the case of the "all" method). For each of the six different ways of generating τ_i described previously, I vary the choice of method used. So for each of six different ways of generating τ_i, I have six different methods and nine different data generating processes, for a total of $6 \times 6 \times 9 = 324$ different simulations.

The three sets of simulations combined result in 1404+171+324=1899 different simulations. I then repeat for the case with a binary dependent variable. For each of the 1899 simulations, I derive the posterior from the algorithm described in the beginning. Due to computational and time issues, each MCMC is relatively short with a chain length of 2000. For the most part, my parameters are relatively

independent so I am confident that my parameters are mixing well despite such a short chain length and no burn-in period.

3 Results from the Simulations

The simulations show that my model in general does a fairly good job at estimating ICEs, although with very high variance in both the estimates (posterior variance) and the quality of the estimates. Although the simulations produce many results and insights that are noteworthy, I only present a subset of the results that can guide researchers on the best practices and methods for estimating ICEs. The rest of the results from the simulations appear in the appendix. The general insights from the simulations are:

1. The model generally performs well in recovering ICEs and other causal estimands. Predictive mean matching generally outperforms all the other matching methods.

Figure 1 shows the results from the first set of simulations comparing the model using the different matching methods with different specifications and sample sizes. The metric here is the average ICE posterior mean bias, which is equivalent to the ATE posterior mean bias. A method or specification is judged by how close its posterior mean bias is to zero. In the top right quadrant with a linear outcome equation and unconfounded treatment assignment, most of the specifications are spot on in their estimate of the ATE.[17] As the outcome equations become more non-linear, the bias gets bigger, but the specifications on average have very little bias until the outcome equations become very non-linear. Looking across methods, the propensity score subclassification method is probably the most consistent in the sense that difference in bias across specifications is the smallest,[18] but the subclassification method is also the most easily biased. The propensity score matching method seems to be the most varied in performance across specifications and its bias seems to be somewhat larger as well. Although the differences are miniscule, it appears that the predictive mean matching method is the method that is most consistent across specifications and has a relatively low bias.

[17] For each method, the different points refer to different specifications of the number of matches, the number of conditioning variables, or the sample size (denoted by color). In all of these graphs, some points are not shown because they fall outside the general range of most of the specifications.

[18] Another way to put it is that the variance of the bias *across* specifications within the subclassification method is the smallest.

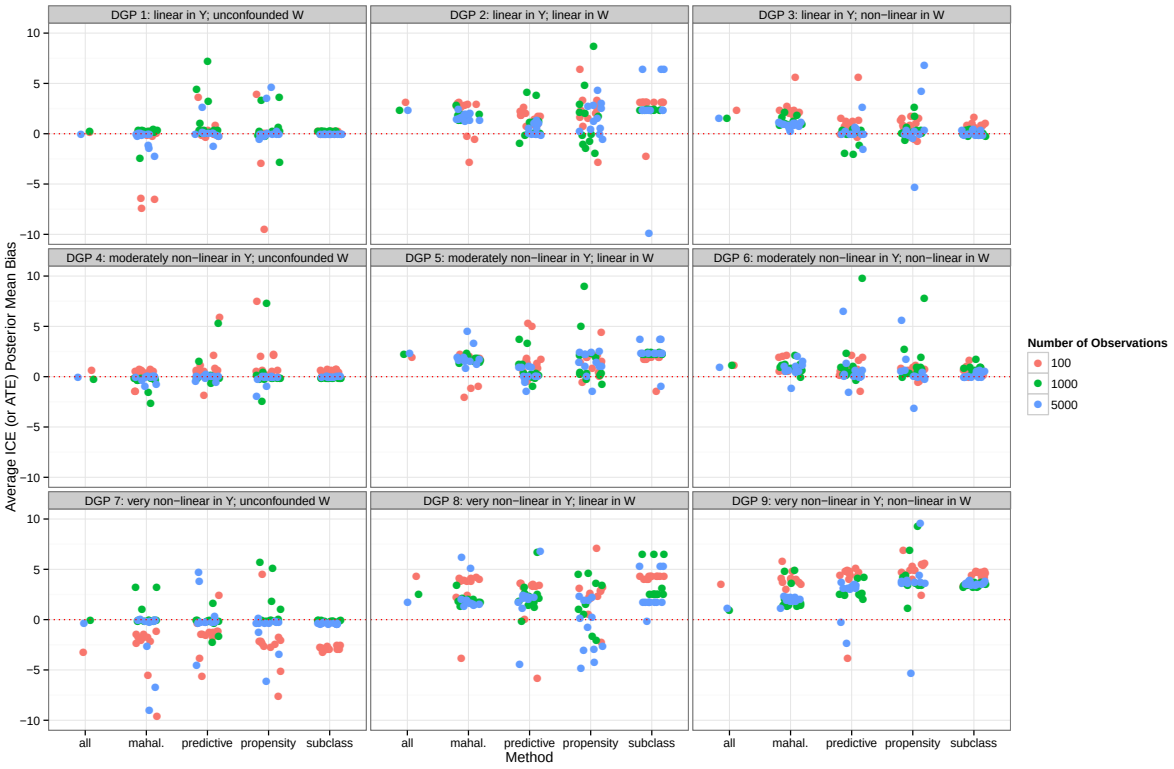

Figure 1: Comparing Average ICE (or ATE) Posterior Mean Bias for the Different Matching Methods (continuous outcome)

Figure 2 shows the results using the average ICE expected error loss as the performance metric. Recall that this metric is analogous to the traditional root mean squared error and gives a sense of both the "bias" and the (posterior) variance of our estimates. A value closer to zero on this metric indicates a better performing method. One can see clearly that the mahalanobis and predictive mean matching methods almost always outperform the other matching methods. Given that the posterior mean bias was similar across the methods, this suggests that mahalanobis and predictive mean matching generally produce more precise estimates with smaller posterior variance. As expected, larger sample sizes also produce estimates with smaller expected error loss.

With a smaller posterior variance, one should also expect predictive mean matching to perform better on the "power" metric of the proportion of 95% credible intervals not including zero since the credible intervals should be smaller. Figure 3 confirms this result where values closer to one on this metric indicate better performance.

In almost all the different data generating processes, the predictive mean matching method performs

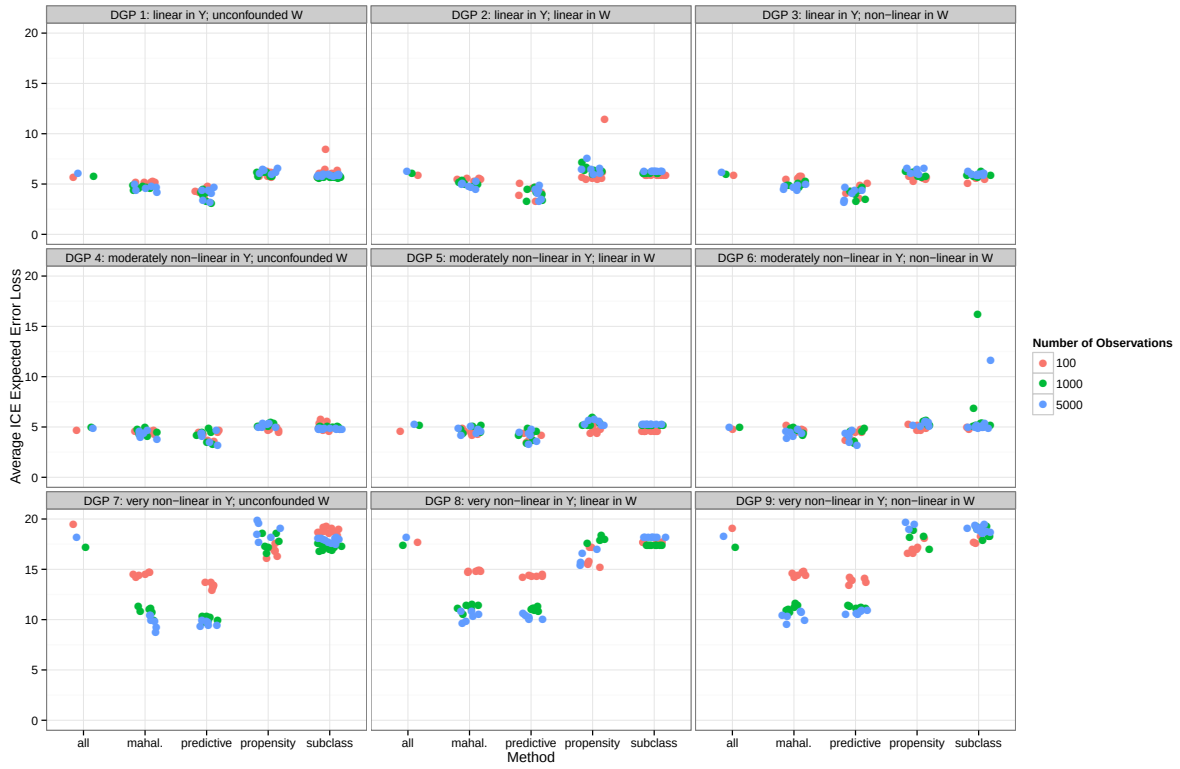

Figure 2: Comparing Average ICE Expected Error Loss for the Different Matching Methods (continuous outcome)

just as well or better than the other methods. However, one thing to note is that for almost every method and specification, the performance on this metric is quite low. The proportion of credible intervals that does not include zero never exceeds 0.5, despite the fact that all the true τ_i are not equal to zero. This result, although undesirable, is expected since the matching methods use a finite and often small number of donor observations, so the posterior variance on the estimate of τ_i is quite high and the credible intervals are quite large. However, the "power" does improve as the actual τ_i get larger. Recall that in traditional methods, the power of a method increases as the effect size gets larger. Figure 4 shows the proportion of 95% credible intervals including zero as a function of the different τ_i distributions.

When the mean of the τ_i distribution is high (at 20), then the "power" is actually quite high for many of the matching methods. Even with a low mean and a high standard deviation, some of the τ_i will be high and so the "power" increases. Drawing τ_i from the mixture distribution of both large and small effects can also increase power relative to only drawing from smaller effects. Thus, although the matching imputation method that I suggest frequently cannot detect small effects, it can do quite well with larger effects. Also, although I use "power" as one metric of judging the methods, the typical Bayesian model

Figure 3: Comparing ICE "Power" for the Different Matching Methods (continuous outcome)

is not as concerned with "power" and hypothesis testing, but instead on whether the credible intervals are properly calibrated and whether the intervals accurately reflect our degree of uncertainty.

Even though my model's 95% credible intervals are quite large, they have the desirable property of being very close to properly calibrated most of the time. Simply put, the large credible intervals have proper "coverage". Figure 5 shows this result. Since I am using 95% credible intervals, a method or specification is said to be properly calibrated if the calibration coverage is at 0.95.

Figure 5 that most of the specifications are around the 0.95 range. As the data generating process becomes more non-linear, the calibration coverage becomes worse, but it is still usually greater than 0.8. The calibration also improves with larger sample sizes. It does not appear that any particular method performs significantly better or worse on this metric. The results here suggest that the credible intervals from the matching methods give about the correct amount of estimation uncertainty.

The results from the first set of simulations confirm that my model performs as well as one might expected, although not perfectly. Predictive mean matching seems to perform as well or better than

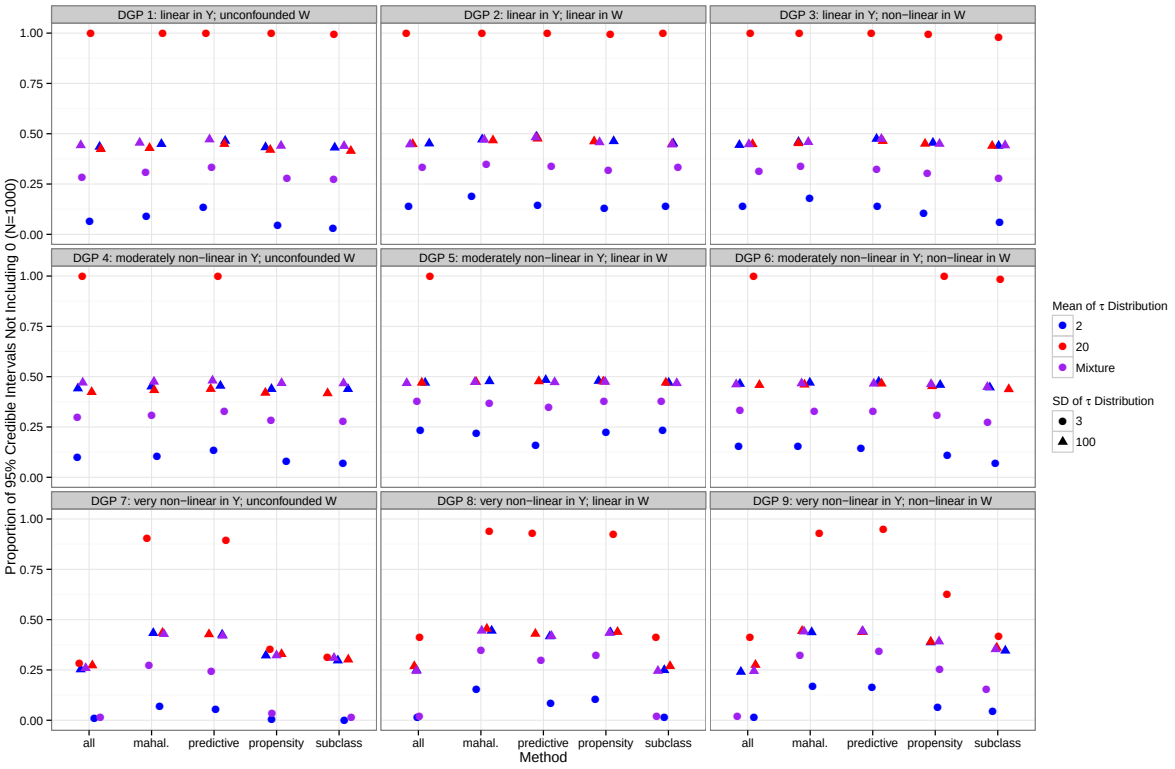

Figure 4: Comparing ICE "Power" with Different τ_i Distributions (continuous outcome)

any other matching method in recovering the causal estimands of interest. Additional results in the appendix lead to a similar conclusion. While more research is needed into assessing why predictive matching performs better, I can offer at least one possible explanation. Recall that the point of the model and estimating ICEs is to impute the missing potential outcome for each observation. The basic idea of predictive mean matching is to first run a regression using all the data for one treatment group to predict the missing potential outcomes for the other treatment groups. The coefficients from that regression are used to match on the predicted means to form a donor pool for a missing potential outcome. This iterative process actually imputes twice; once to get a rough mean to determine the donor pool and again to actually impute from the donor pool. The objective of predictive mean matching most closely resembles the objective of estimating ICEs in imputing potential outcomes and the two-step iterative process allows for improvements in the imputations. In a sense, I propose a causal framework in my model but use a data mining/machine learning type algorithm in practice. This allows me to achieve optimal results while retaining the principle of modeling the causal process. This may explain why predictive mean matching in my model performs best in practice.

Figure 5: Comparing ICE Calibration Coverage for the Different Matching Methods (continuous outcome)

2. Regression imputation works well for estimating average effects. It offers more precise estimates for individual and average effects, but the uncertainty does not accurately reflect the correct uncertainty in estimating the ICEs. Compared to regression imputation, predictive mean matching in my model gives estimates that are almost as good and the uncertainty estimates are correct.

The simplest and most straightforward way to estimate ICEs is by imputing the missing potential outcomes with a regression model. Regression imputation takes the regression model of Y on W and X and imputes using the fitted values from the regression coefficients, simply changing the treatment assignment indicator to the missing one. I compare this simple method of (Bayesian) regression imputation with my Bayesian imputation model using predictive matching, which I showed was the best performing matching method.

Figure 6 shows the posterior mean bias of regression imputation versus predictive mean matching. Unsurprisingly, regression imputation performs very well in recovering the ATE, a quantity that it was

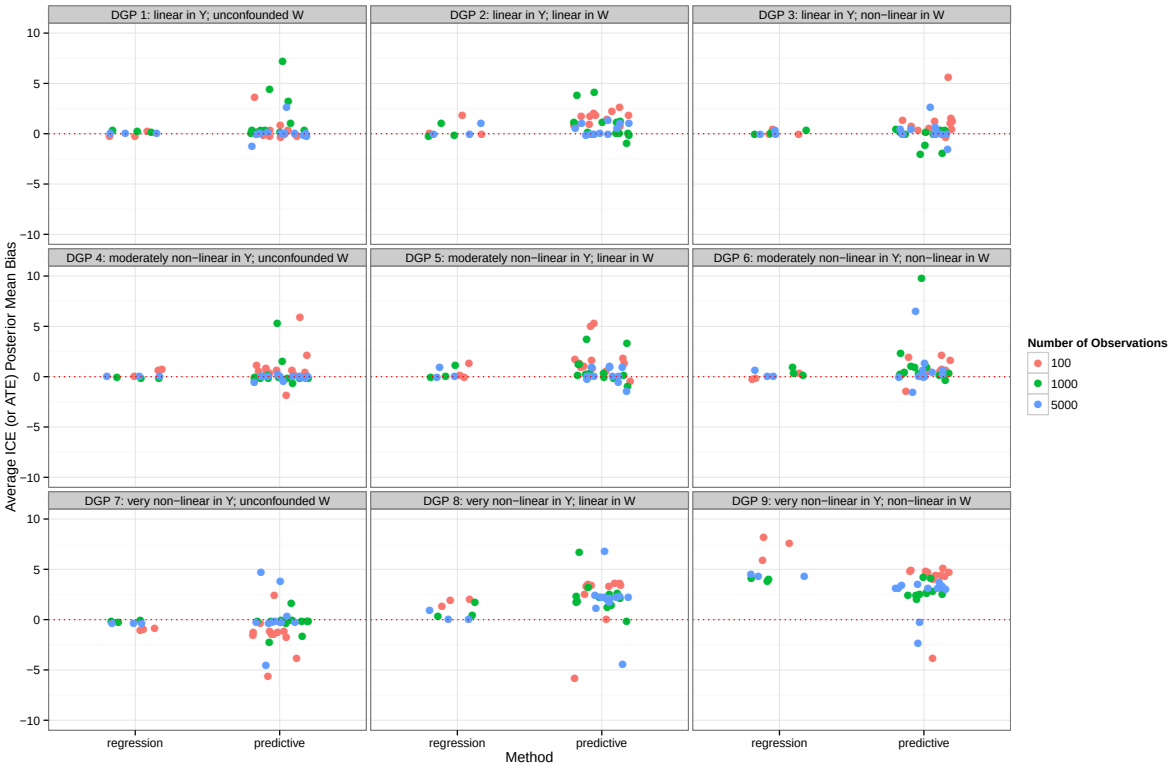

Figure 6: Comparing Average ICE (or ATE) Posterior Mean Bias for Regression Imputation versus Predictive Mean Matching Model (continuous outcome)

designed to capture. It consistently gets it right over various specifications. Predictive mean matching performs almost as well, although it is less consistent across various specifications. As the data generating process becomes more non-linear, the functional form for both methods is incorrect and the estimates become less accurate. To confirm that both methods perform about as well in estimating ICEs, I look at "point estimation" for both methods in Figure 7. In Figure 7, I use the specification with sample size 100 and 7 conditioning variables for both methods and 25 matches for the predictive mean matching. For each method, I take the posterior means for each ICE and take the absolute differences between the posterior means and each true τ_i. This captures how far off each method is for each ICE. I then difference these differences to capture the relative performance of each method for each ICE. Each point on the graph represents the difference in difference for each ICE, so there should be 100 points for each data generating process. A point above the zero line indicates that the "point estimate" for predictive mean matching is closer to the true ICE for that specific ICE and a point below the zero line indicates that the "point estimate" for regression imputation is closer. The red points indicate the median on the difference-in-difference scale. It appears that there is no specific pattern to the distribution of differences-in-differences. Most of the points seem randomly distributed around the zero line, which indicates that

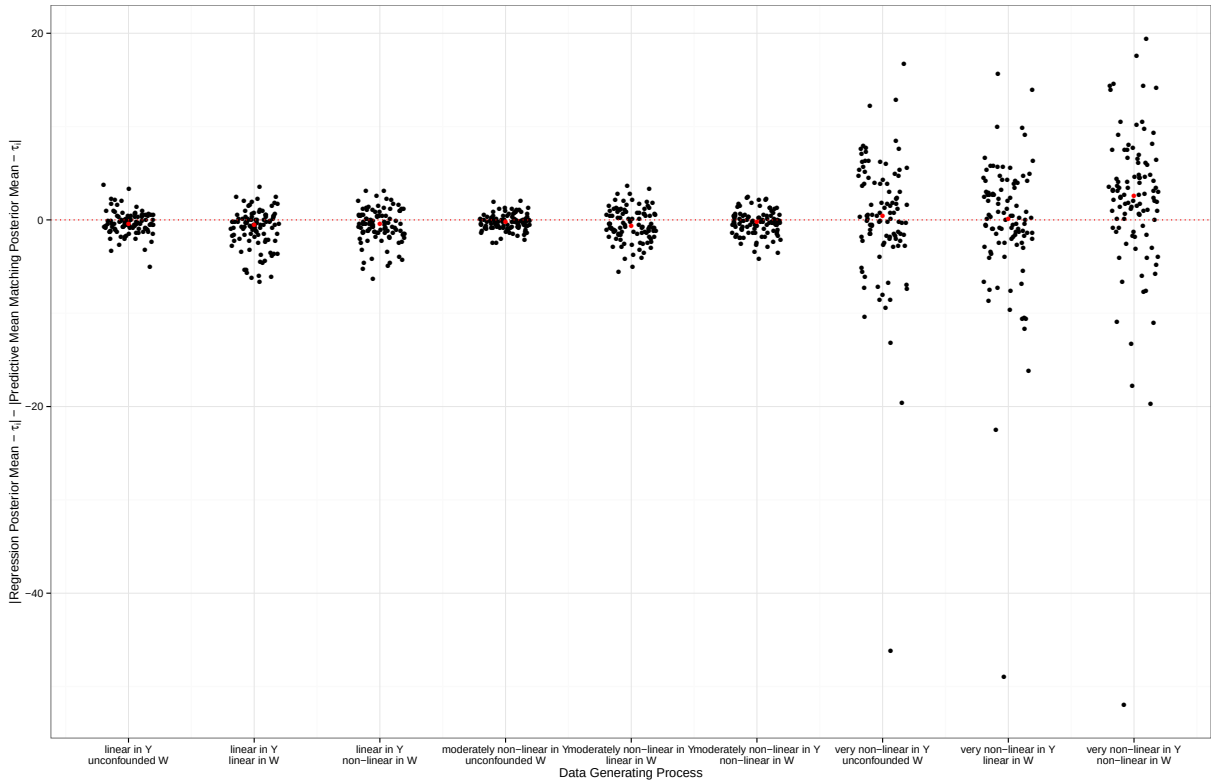

Figure 7: Comparing the Absolute Differences Between Posterior Means and the True ICE for Regression Imputation versus Predictive Mean Matching Model (continuous outcome)

for some observations, regression imputation does better and for others, predictive mean matching does better. For the very non-linear generating equations, the differences become more spread out and outliers occur more frequently. Nevertheless, it appears that both regression imputation and predictive mean matching perform similarly on "point estimation" of ICEs.

Although the performance on point estimation is similar for both regression and matching, regression imputation gives posteriors that have smaller variances, as shown in Figure 8, which plots the results of average ICE expected error loss. The expected error loss is generally smaller for regression imputation. This is also unsurprising since regression imputation makes an added assumption of modeling only the average. This added assumption allows for more precise estimates and subsequently more "power", as Figure 9 demonstrates. Figure 9 shows that regression imputation is able to detect $\tau_i \neq 0$ at a much higher rate than predictive mean matching. By modeling only the average effect and imputing from the model, regression imputation results in much smaller posterior variance. Recall that in regression imputation, the model uses all the observations to model the ATE, which results in a relatively small posterior variance for the ATE. This posterior is used directly in the imputation and posterior for each

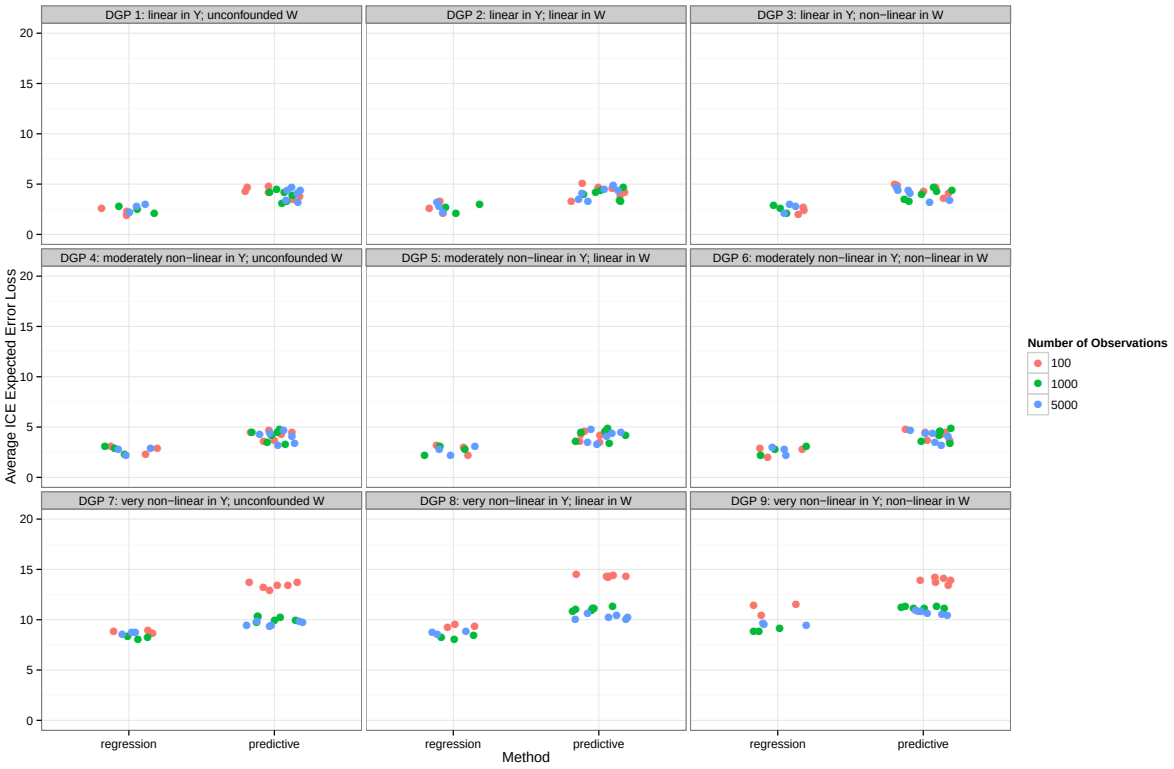

Figure 8: Comparing Average ICE Expected Error Loss for Regression Imputation versus Predictive Mean Matching Model (continuous outcome)

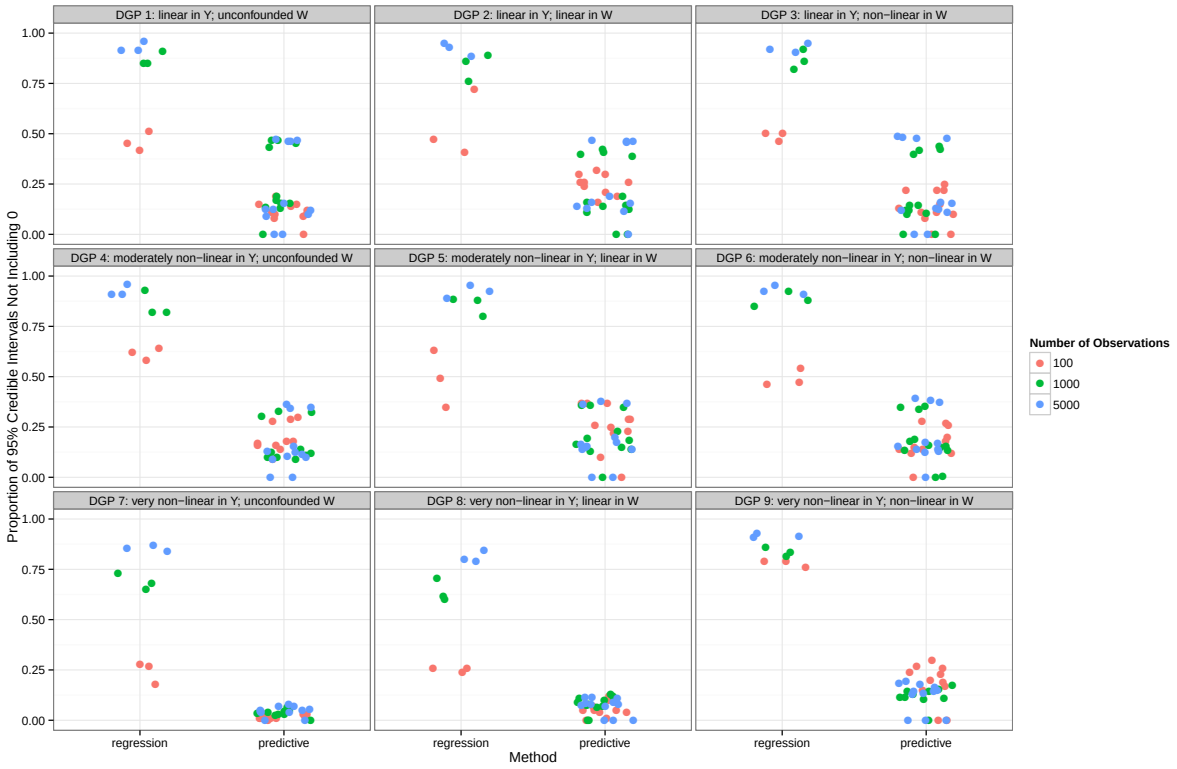

Figure 9: Comparing ICE "Power" for Regression Imputation versus Predictive Mean Matching Model (continuous outcome)

ICE, so the width of the posterior credible interval for the ICE is the same as the width of the credible interval for the ATE. Contrast this with my imputation model with predictive mean matching, where the width of the credible interval for an ICE is derived from matching on a smaller set of donor pool observations. It is straightforward to see that regression imputation results in smaller credible intervals, which in turn decreases the probability of zero appearing in the credible interval and thus more "power".

Given that regression imputation produces estimates that are just as "correct" as my imputation model with predictive mean matching with smaller credible intervals and more power, why would one not use regression imputation for estimating ICEs? It turns out that the credible intervals are actually too small, which is unsurprising since they are credible intervals designed for the ATE rather than ICEs. Regression imputation is very poorly calibrated, and the uncertainty that is reflected by the posterior variance is incorrect for ICEs. Figure 10 shows the results of the calibration coverage for the 95% credible intervals. While approximately 95% of the 95% credible intervals cover the true τ_i for predictive mean matching, most of the time less than 50% of the 95% credible intervals do so for regression imputation. The smaller credible intervals lead to incorrect inferences more than half the time.

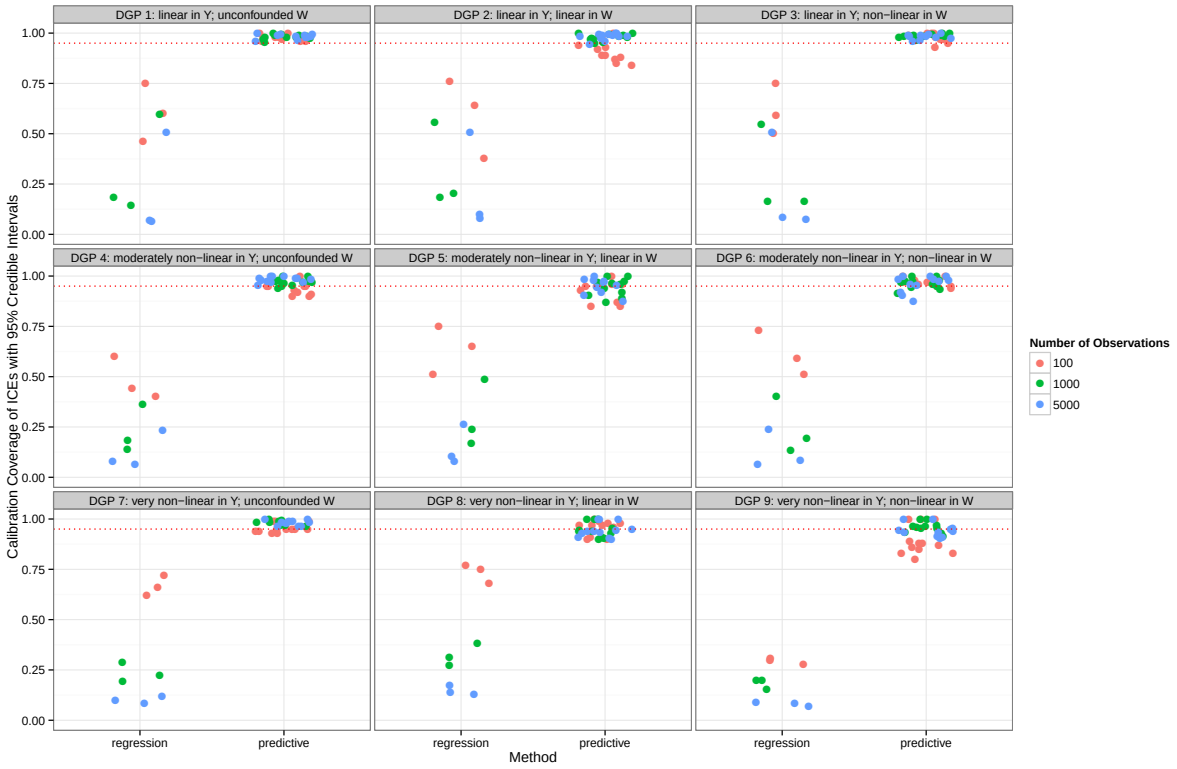

Figure 10: Comparing ICE Calibration Coverage for Regression Imputation versus Predictive Mean Matching Model (continuous outcome)

Figure 11 shows the different calibration coverages when drawing τ_i from different distributions. While predictive mean matching is accurately calibrated regardless of the distribution of the true τ_i, regression imputation is also very poorly calibrated regardless of the distribution of the true τ_i. There appears to be a general pattern that the calibration for regression imputation is better when the τ_i are more spread out (higher standard deviation of the τ_i distribution). One possible explanation is that when the τ_i are more spread, the posterior variance for the ATE is larger and so the credible intervals for the ICE are also larger and thus will include the true τ_i a greater proportion of the time. Nevertheless, it is clear that while regression imputation does just as well as my matching imputation model in point estimation of the ICEs, it is a poor technique for estimating the uncertainty of the ICEs and should be used only for modeling averages rather than individual effects. The typical method of imputing from a regression model is incorrect when looking at individuals.

Figure 11: Comparing ICE Calibration Coverage with Different τ_i Distributions (continuous outcome)

3. There appears to be no discernible difference in the number of X variables to condition on as long you condition on all (or almost all) confounders.

Although more simulations are needed to fully test the effect of omitting or including conditioning variables, it appears that as long as one conditions on all or close to all of the confounders, adding extra prognostic variables to the conditioning set does not result in drastic improvements. Figure 12 shows the results of average ICE expected error loss across all the matching methods with 0, 5, 7, and ten conditioning variables. Recall that for the specifications with confounded treatment assignment, 7 is the correct number of confounders. Conditioning on ten X variables means conditioning on all the confounders and all the prognostic variables. The results suggest that there are no discernible differences in performance when conditioning on 5, 7, or 10 confounders across all the different data generating processes. This suggests that as long as one controls for approximately the correct confounders, the results should be quite stable. However, I do not test the effect of omitting very important versus less important confounders or including or excluding very important prognostic variables. Future research should look into these questions in more detail.

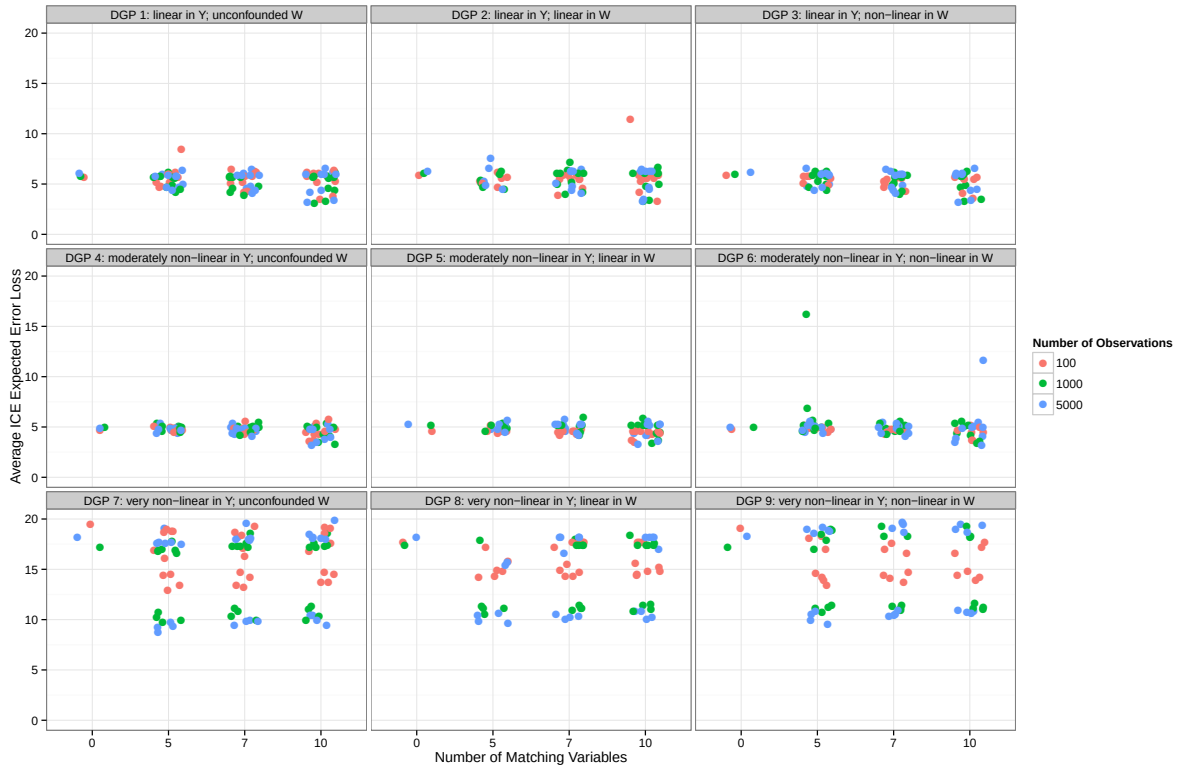

Figure 12: Comparing Average ICE Expected Error Loss for Different Conditioning Sets (continuous outcome)

4. The optimal number of matches to use is dependent on the data generating process, although one should not use a very small number or a very large number of matches. A random number of matches does not seem to provide a huge improvement compared to a fixed number of matches.

In typical matching analyses, there is a bias-variance tradeoff between using too few versus too many matches. When using a small number of matches, bias is small since only high quality matches are used, but variance is large with such a small donor pool. When using a large number of matches, variance is smaller but lower quality matches are included in the donor pool, which may increase bias. There is a slightly different story when using matching to estimate ICEs in my model. Figure 13 shows the posterior mean bias from using different sizes of donor pools across the different matching specifications.[19] Recall that in my specifications, a small number of matches is 2, medium is 10, large is 25, and random is a randomly drawn integer between 2 and 25 for each iteration of the algorithm. For a very small number of matches, the posterior mean bias is quite unstable across various specifications. Using a medium or

[19] Each point represents a different specification of matching method and number of conditioning variables. The subclassification method is not included in these results.

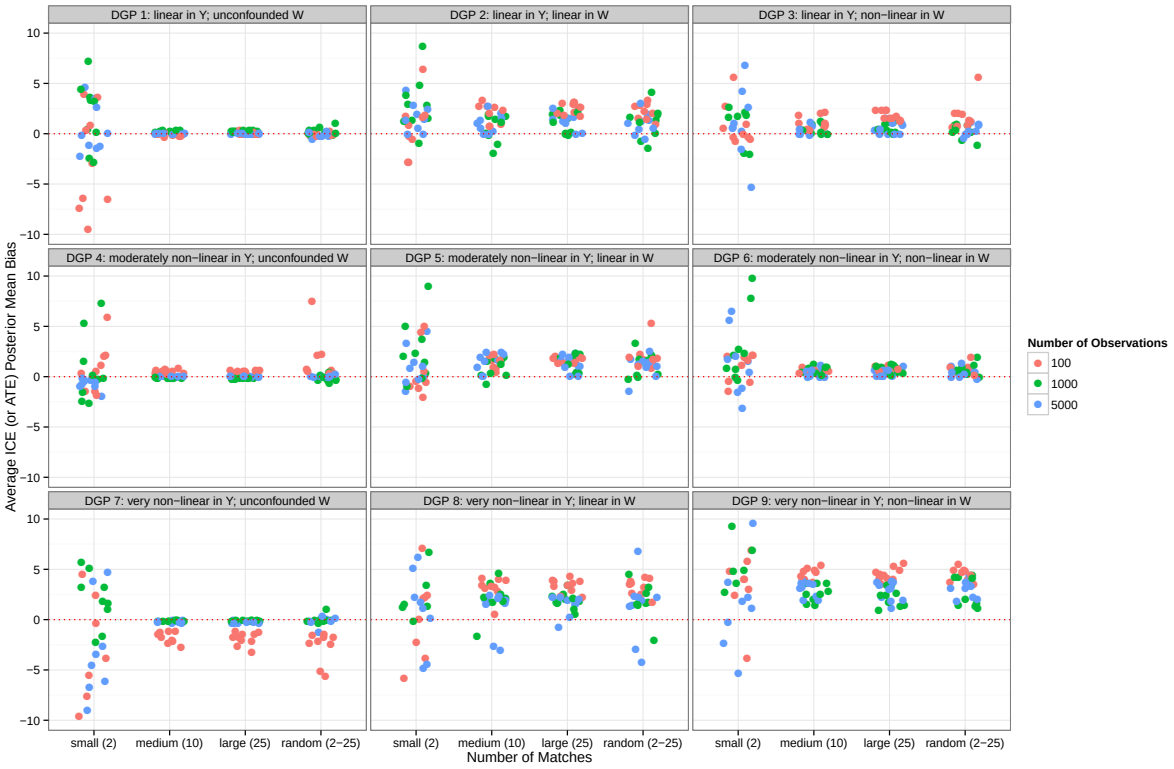

Figure 13: Comparing Average ICE (or ATE) Posterior Mean Bias for Different Numbers of Matches (continuous outcome)

large number of matches seems to give better and more consistent results. There seems to be no benefit to using a random versus fixed number of matches. Figure 14 shows the results of average ICE expected error loss, which takes into account posterior variance. When using only two matches, there is a large error, which represents both poor "point estimates" and large posterior variance. Using a slightly larger number of matches shrinks the expected error significantly. Also, using a random number of matches increases the variance of the results without a large increase in posterior mean bias.

While one can look at the previous results and conclude that larger numbers of matches are better, using 25 matches is large for a sample size of 100 but quite small for a sample size of 5000. I further test the idea of optimal number of matches by looking at the number of matches as a percentage of the number of observations in the smaller treatment group. Figure 15 shows the posterior mean bias for the different match percentages using a specification with predictive mean matching on 7 confounders with sample size of 1000. As the match percentage (or equivalently the number of matches) increases, the posterior mean bias also tends to increase, which suggests that larger donor pools are incorporating poorer quality matches and inducing "bias". There does not appear to be an optimal match percentage

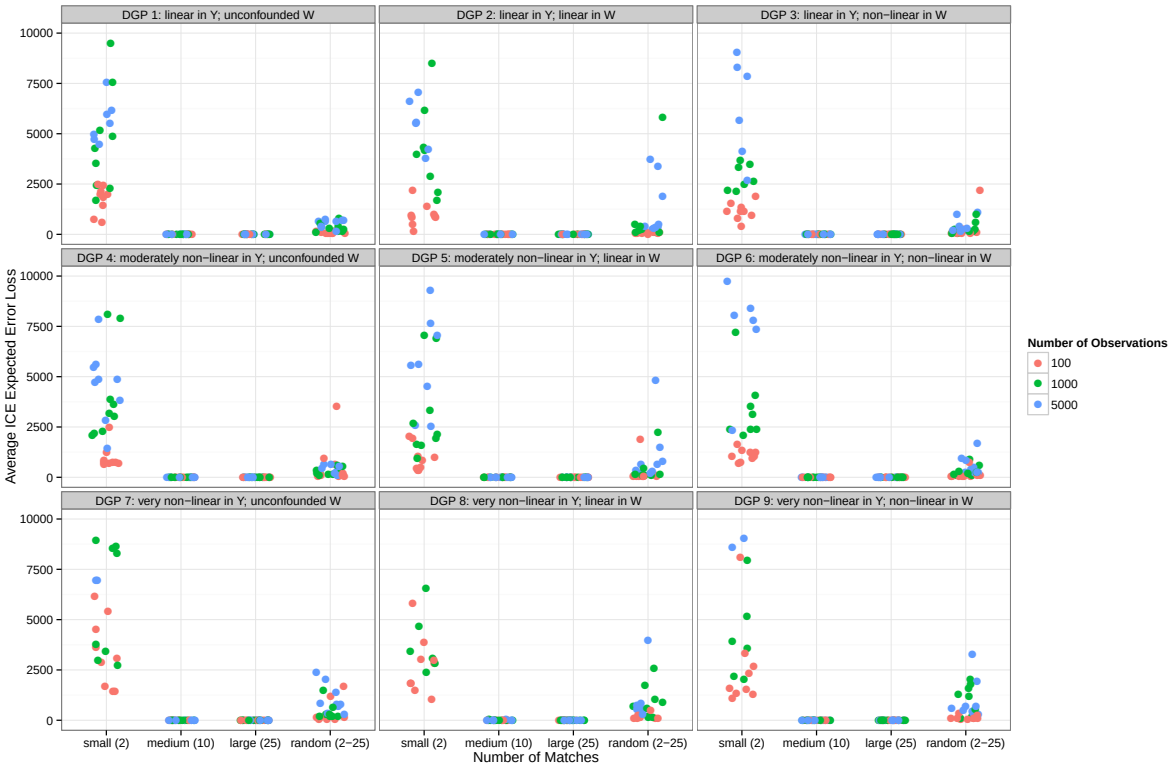

Figure 14: Comparing Average ICE Expected Error Loss for Different Numbers of Matches (continuous outcome)

for all data generating processes, although I suggest that 10% of the smaller treatment arm seems to be a good number to use that consistently gives decent results. The results on other metrics (in the appendix) also confirm that there is no optimal number and 10% seems to work well.

4 Conclusion

The simulation results I have presented here and in the appendix are only the tip of the iceberg for testing my model and the different specifications. I have tried to test my model and compared it to imputation from regression, which is the simplest and most widely used way to estimate and predict individual effects. I conclude that predictive mean matching performs the best out of the matching methods I propose. I also show that both regression imputation and predictive mean matching do fairly well in "point estimation" of the ICEs, but regression imputation gives uncertainty estimates that are wildly incorrect whereas my model is properly calibrated. For practical use, I suggest using predictive mean matching with a fixed donor pool size of approximately 10% of the smaller treatment arm, conditioning

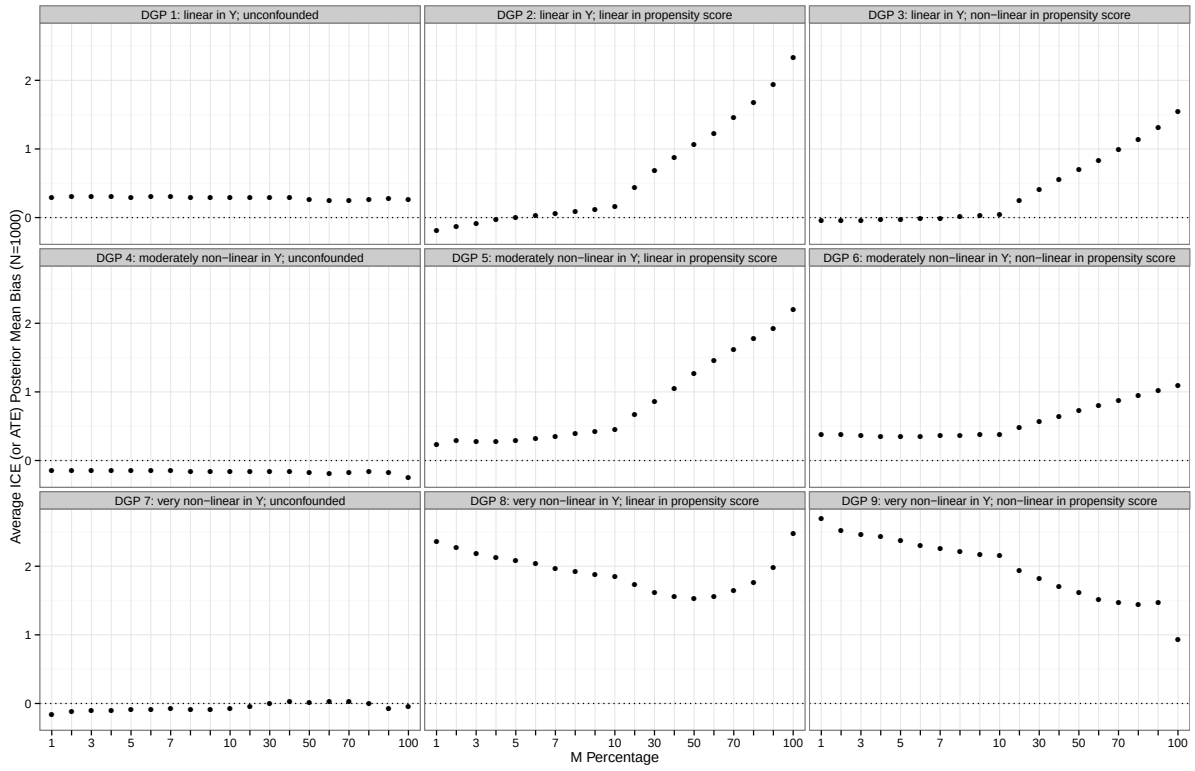

Figure 15: Comparing Average ICE (or ATE) Posterior Mean Bias for Different Match Percentages (continuous outcome)

on all observed confounders.

References

An, Weihua. 2010. "Bayesian Propensity Score Estimators: Incorporating Uncertainties in Propensity Scores into Causal Inference." *Sociological Methodology* 40(1):151–189.

Frölich, Markus. 2007. "Propensity Score Matching without Conditional Independence Assumption with an Application to the Gender Wage Gap in the United Kingdom." *Econometrics Journal* 10:359–407.

Hainmueller, Jens. 2012. "Entropy Balancing for Causal Effects: A Multivariate Reweighting Method to Produce Balanced Samples in Observational Studies." *Political Analysis* 20(1):25–46.

Rosenbaum, Paul R. and Donald B. Rubin. 1984. "Reducing Bias in Observational Studies Using Subclassification on the Propensity Score." *Journal of the American Statistical Association* 79(387):516–524.

Rosenbaum, Paul R. and Donald B. Rubin. 1985. "The Bias Due to Incomplete Matching." *Biometrics* 41(1):103–116.

Rubin, Donald B. 1980. "Bias Reduction Using Mahalanobis-Metric Matching." *Biometrics* 36(2):293–298.

Rubin, Donald B. 2001. "Using Propensity Scores to Help Design Observational Studies: Application to the Tobacco Litigation." *Health Services & Outcomes Research Methodology* 2:169–188.