CBPS and Entropy Balancing - A simulation Study

Chansoo Song

## Motivation:

A key challenge in the application of propensity scores for matching is that the propensity score is unknown and must be estimated. To make matters worse, slight misspecification of the propensity score model can lead to substantial biases in treatment effects. This has led to researchers iteratively re-estimating the propensity score model, subsequently checking the resulting covariate balance, then repeating over and over until they are satisfied. Imai et al (2008) calls this the ‘propensity score tautology’: the estimated propensity score is appropriate if it balances covariates.

In this simulation study, I analyze two approaches that seek to bypass this ‘propensity score tautology’: Covariate Balancing Propensity Score and Entropy Balancing. Each method obviates the need for iteratively re-estimating the propensity score model and checking balance on the covariate moments. That is, a single model is used to estimate both the treatment assignment mechanism and the covariate balancing weights.

## Assumptions:

#### Matching:

In an observational study setting where the confounding covariates are known and measured, we may use matching methods to ensure that there is sufficient overlap and balance on these covariates. Then, we can estimate the treatment effect using a simple difference in means or regression methods.

Overlap is important because we want to make sure that for each treated or control subject in the study, there exists an empirical counterfactual (this criteria varies depending on the estimand of interest, i.e. to estimate the ATT it is sufficient to have empirical counterfactuals for just the treated subjects in the study). Balance on the covariates is important because imbalance would force us to rely more on the correct functional form of the model.

There are many different matching methods, but the driving principle is to identify observations that are “most similar”, based on some distance metric. Methods include K-nearest-neighbor, caliper-matching, kernel-matching, Mahalanobis matching, Genetic Matching, Optimal Matching.

#### Propensity Scores:

A propensity score is a one-number summary of the covariates. Rosenbaum and Rubin (1983) define the propensity score for participant i as the conditional probability of treatment assignment given a vector of observed covariates: . The most common traditional approaches to estimating the propensity score are logistic regression and probit regression.

If strong ignorability holds after conditioning on the propensity score, that is:

Then we may obtain an unbiased estimate of the treatment effect by either matching or weighting using just the propensity score instead of the vector of covariates.

## Describe the designs / estimators:

#### Covariate Balancing Propensity Score (CBPS):

The CBPS exploits the dual characteristics of the propensity score as a covariate balancing score and the conditional probability of treatment assignment (Imai and Ratkovic (2012)).

First, consider a commonly used model for estimating propensity scores: logistic regression:

We typically estimate the unknown parameters by maximum likelihood, i.e. maximizing the log-likelihood function:

Differentiating with respect to , we get:

Imai and Ratkovic emphasize that the equation above can also be interpreted as the condition that balances a particular function of covariates, i.e. the first derivative of . Then, they operationalize the covariate balancing property by using inverse propensity score weighting:

where , a function of specified by the researcher. Setting gives more weights to covariates that are predictive of treatmeent assignment according to the logistic regression propensity score model. But so long as the expectaion exists, the equation must hold for any choice of f(.). For example, setting ensures the first moment of each covariate is balanced. Setting ensures the first and second moment of each covariate is balanced.

Imai uses “the moment conditions based on the covariate balancing property under the GMM or EL framework”.

#### Entropy Balancing:

Entropy balancing similarly involves a reweighting scheme that directly incorporates covariate balance into the weight function that is applied to the sample units (Heinmueller 2015). To do this, entropy balancing searches for a set of weights that satisfies the balance constraints, while trying to keep the distribution of weights as uniform as possible (i.e. minimizing the divergence of distribution of weights from a uniform distribution). Thus, entropy balancing (1) allows us to obtain a high degree of covariate balance (using balance constraints that can involve the first, second, and possibly higher moments of the covariate distributions as well as interactions). And (2) allows for a more flexible reweighting scheme that seeks to retain as much information as possible. For example, nearest neighbor matching may discard subjects that are not matched (i.e. set weight equal to 0).

Consider the reweighting scheme to estimate the Average Treatement Effect on the Treated (ATT). We would want to estimate the counterfactual mean by:

where is a weight for each control unit.

The weights are chosen by the following reweighting scheme:

where is a distance metric and describes a set of R balance constraints imposed on the covariate moments of the reweighted control group.

Minimize subject to the balance and normalizing constraints:

with

and for all such that .

#### Comparison and implementation:

Both methods may be used to estimate either the ATT, ATC, or the ATT.

The two methods are very similar. The key difference is that entropy balancing bypasses the ‘propensity score tautology’ by ignoring the propensity score model estimation step. Instead, entropy balancing seeks to find weights that achieve the best balance subject to a constraint that seeks to retain as much information in the data as possible. In contrast, covariate balancing directly models the propensity score and constructs weights simultaneously.

I use the ‘ebal’ and ‘CBPS’ packages in R to implement Entropy Balancing and Covariate Balancing Propensity Score, respectively.

## Simulation Set Up:

#### Features:

In this section, I examine whether the CBPS or Entropy Balancing methods improve upon the performance of a baseline approach to both (1) achieving balanced covariates and (2) estimating the treatment effect. The baseline approach estimates include (1) propensity scores using logistic regression and matches using 1-1 matching with replacement and (2) mahalanobis matching.

Though ignorability may be the most crucial assumption, I assume that all of the confounders are known to the researcher for all simulations. I believe that testing the sensitivity to the ignorability assumption would be more interesting when comparing propensity score and matching methods to much more different approaches to causal inference. All of the models that I am testing similarly require ignorability to be satisfied. However, the authors of CBPS and EB claim that these models are less dependent on correctly specifying the propensity score and outcome models, compared to traditional propensity score approaches. Further, these models do a better job of achieving good balance on the covariates. Thus, I am concerned with (1) reliance on the correct specification of the propensity score model, (2) reliance on the correct specification of the outcome model, and (3) reliance on the ellipsoidal shape of covariate distributions.

#### Estimand:

The estimand of interest is the ATT: Average Effect of Treatment on the Treated. The ATT tells us how much the treatment affected the group of subjects that receieved treatment. We estimate the ATT by comparing the observed outcomes to the outcomes that we would have measured had this group of subjects not received treatment. But since we are not able to observe this counterfactual state, we match each individual in this group of treated subjects to a control subject. In general, since the treatment group usually represents a specific subset of the general population, we would expect that the range of covariate measures on the treated are a subset of the range of covariate measures on the general population as well. Then, since we would have overlap, it makes sense to try to use matching and propensity score techniques to achieve good balance. On the otherhand, in general, estimating the effect on the control would be a much greater challenge. We would face the challenge of matching a control subject with a treated subject. Though there may be exceptions, most studies are concerned with a specific treatment on a specific group of people. There would be substantial overlap issues that would prevent reliable extrapolation of inferences based on this specific subset to draw conclusions about how an entirely different groups of people would have responded to treatment.

#### DGPs and other simulation parameters:

I consider eight data generating processes:

|  |  |  |  |
| --- | --- | --- | --- |
| DGP | Unequal Variances | Incl. Count Covariates | Linear Propensity Score Model |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 0 | 0 |
| 3 | 1 | 0 | 1 |
| 4 | 1 | 0 | 0 |
| 5 | 0 | 1 | 1 |
| 6 | 0 | 1 | 0 |
| 7 | 1 | 1 | 1 |
| 8 | 1 | 1 | 0 |

1. Standard Normally Distributed Covariates: The pre-treatment covariates are four independent and identically distributed random variables following a standard normal distribution. The true propensity score model is a logistic regression whose linear predictor is a linear transform of the pre-treatment covariates.
2. Standard Normally Distributed Covariates (non-linear propensity score model): The pre-treatment covariates are the same as Simulation #1; however, the true propensity score model is a logistic regression whose linear predictor are non-linear transforms of the pre-treatment covariates:
3. Normally Distributed Covariates with Unequal Variances Same as (1) except the variances of the pre-treatment covariates are no longer equal. That is, the covariates are no longer identically standard normal, but have variances: (0.5, 0.9, 1.1, and 1.2), respectively.
4. Normally Distributed Covariates with Unequal Variances (non-linear propensity score model): Same as (2) except the variances of the pre-treatment covariates are no longer equal. That is, the covariates are no longer identically standard normal, but have variances: (0.5, 0.9, 1.1, and 1.2), respectively.
5. Standard Normally Distributed Covariates + 3 count covariates: The pre-treatment covariates consist of four independent and identically distributed random variables following a standard normal distribution, a random variable following a poisson distribution with , the negative values of a random variable following a binomial distribution with and , and a random variable following a chi-squared distribution with .
6. Standard Normally Distributed Covariates + 3 count covariates (non-linear propensity score model): The pre-treatment covariates are the same as Simulation (5); however, the true propensity score model is a logistic regression whose linear predictor are non-linear transforms of the pre-treatment covariates:
7. Normally Distributed Covariates with unequal variances + 3 count covariates: Same as (5) except the variances of the pre-treatment covariates are no longer equal. That is, the covariates are no longer identically standard normal, but have variances: (0.5, 0.9, 1.1, and 1.2), respectively.
8. Normally Distributed Covariates with unequal variances + 3 count covariates (non-linear propensity score model): Same as (5) except the variances of the pre-treatment covariates are no longer equal. That is, the covariates are no longer identically standard normal, but have variances: (0.5, 0.9, 1.1, and 1.2), respectively.

I run each DGP twice, for a total of 16 simulations. For the first set of eight simulations, the true outcome model is a linear regression with the pre-treatment covariates as predictors. For the second set of eight simulations, the true outcome model is a linear regression with non-linear transformations of the pre-treatment covariates as predictors.

## Simulations:

#### Set Up and Initialize:

#### DGP Function (Pre-Treatment Covariates):

#### Generate Researcher Dataset Function:

#### Matching Methods

1. Baseline: Propensity Score using Logistic Regression:
   * Logistic Regression to estimate propensity scores
   * 1-1 nearest neighbor matching using propensity score
   * linear regression using propensity score and covariates
2. Baseline: Mahalanobis Matching

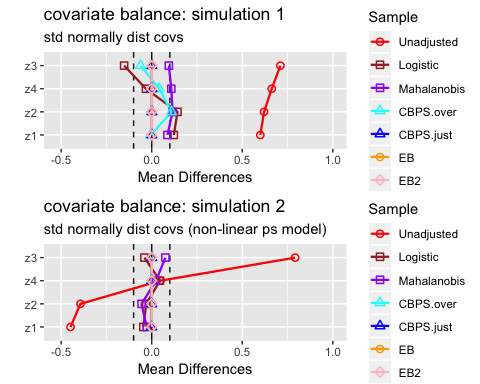
Mahalanobis calculates distance as and it is equivalent to Euclidean matching based on standardized and orthogonalized X. Mahalanobis was intended for use with multivariate normally distributed data. When some covariates exhibit extreme outliers or very skewed distributions, Mahalanobis distance will place less weight on that covariate. On the other hand, a binary variable with a .99 probability of one would have low standard deviation and the Mahalanobis distance would give greater weight to this variable. One way to address these concerns would be to use ar rank-based Mahalanobis distance. For this simulation study, I use the standard Mahalanobis distance.

1. Covariate Balancing Propensity Score
   * Over-identified: combines the propensity score and covariate balancing conditions
2. Covariate Balancing Propensity Score
   * Just-identified: only contains covariate blancing conditions
3. Entropy Balancing

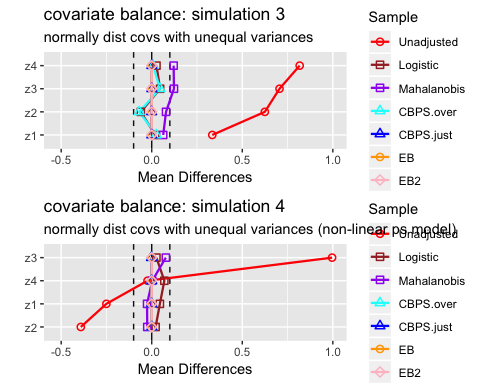
#### Estimate Weights

#### Examine Overlap of Propensity Score and Covariates (Before Matching)

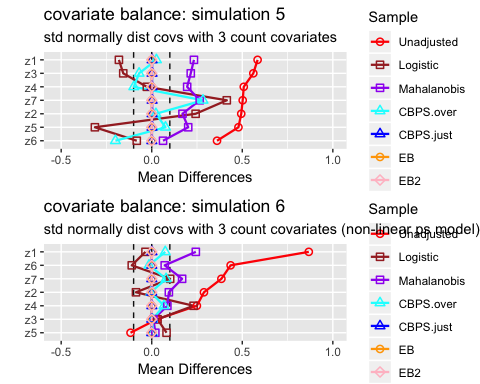
grid.arrange(grobs=love.plots[1:2], ncol=1, nrow = 2)



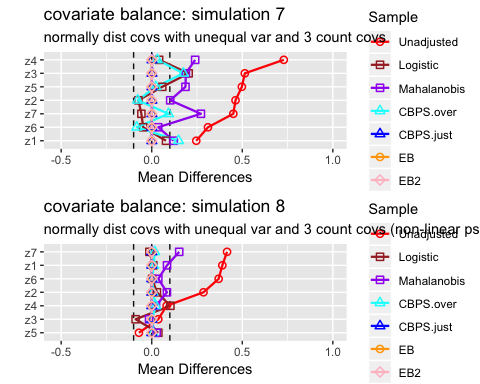
grid.arrange(grobs=love.plots[3:4], ncol=1, nrow = 2)



grid.arrange(grobs=love.plots[5:6], ncol=1, nrow = 2)



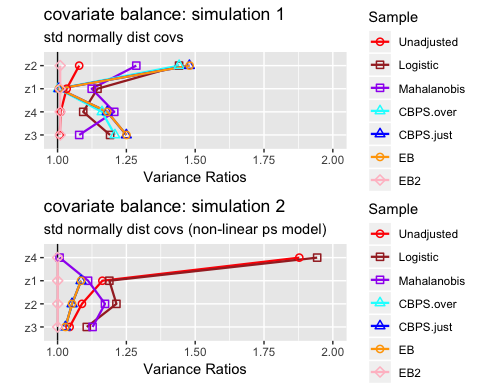
grid.arrange(grobs=love.plots[7:8], ncol=1, nrow = 2)



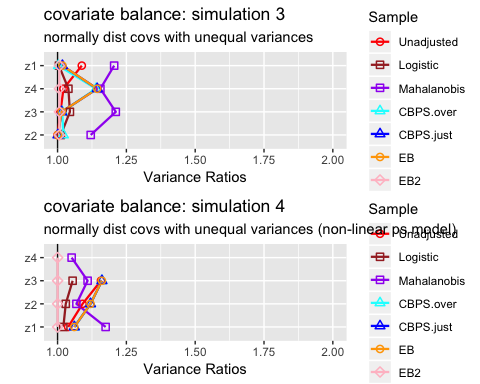
#### Analysis of Mean Differences:

* Entropy balancing shows the best performance with respect to balancing covariate means. For all 8 simulations, the standardized mean difference is approximately zero for all covariates.
* The CBPS just-identified model similarly achieves balanced covariate means, except for simulation #7, where discrete variables are included and the continuous variables have unequal variance.
* The CBPS over-identified model performs significantly better in simulations where the true propensity score model is non-linear and worse in simulations where the true propensity score model is linear. In fact, for 3 of 4 simulations with a non-linear true propensity score model, the CBPS over-identified model achieves a 0 mean difference for all covariates. When the true propensity score model is linear, the covariates’ mean differences are very similar to the covariates’ mean differences under the logistic model. Remember that the over-identified model combines the propensity score and covariate balancing conditions whereas the just-identified model only contains covariate balancing conditions. It seems likely that when the true propensity score model is linear in the covariates, the propensity score condition dominates the covariate balancing conditions, so the CBPS over-identified model’s performance resembles the logistic baseline model’s results. For the simulations with a non-linear propensity score model, the propensity score condition no longer dominates, so the CBPS over-identified model’s performance resembles the just-identified model’s results.
* The logistic regression and mahalanobis matching methods show strong performance in simulation 1, where the pre-treatment covariates have standard normal distributions. Performance appears to weaken when variances are not equal and when the true propensity score model is non-linear.

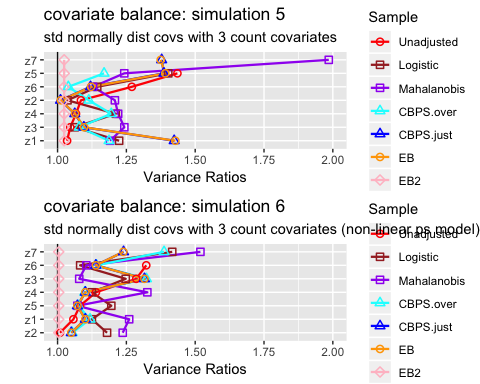
grid.arrange(grobs=love.plots.var[1:2], ncol=1, nrow = 2)



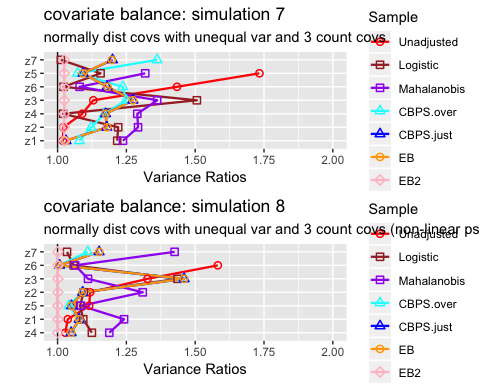
grid.arrange(grobs=love.plots.var[3:4], ncol=1, nrow = 2)



grid.arrange(grobs=love.plots.var[5:6], ncol=1, nrow = 2)



grid.arrange(grobs=love.plots.var[7:8], ncol=1, nrow = 2)



#### Analysis of Variance Ratios:

* The Entropy Balancing model where I have set both first and second moment conditions (EB 2) is the only model that consistently achieves variance ratios = 1. The other models’ ability to obtain similar variances of matched samples across treatment groups is rather sporadic.
* Simulations 3/4 and simulations 7/8 are not discernibly different from simulations 1/2 and simulations 5/6. This suggests that unequal variances in the pre-treatment covariates does not impact variance ratios of matched samples.

#### Sampling Distributions of Estimated Treatment Effect

The block of code below was run on HPC clusters to save time, hence eval=FALSE. I load results of models below for analysis.

#### Analysis of Simulations:

kable(round(lin\_bias1,3),   
 caption='Linear Outcome Model -- Linear Regression: Bias')

Linear Outcome Model – Linear Regression: Bias

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Baseline Logistic | Baseline Mahalanobis | CBPS - Over | CBPS - Just | EB - 1 | EB - 2 |
| sim 1 | 0.006 | 0.011 | 0.008 | 0.008 | 0.008 | 0.009 |
| sim 2 | 0.000 | 0.005 | 0.001 | 0.001 | 0.001 | 0.002 |
| sim 3 | 0.000 | -0.001 | 0.000 | 0.000 | 0.000 | 0.001 |
| sim 4 | -0.001 | -0.008 | -0.003 | -0.003 | -0.003 | -0.009 |
| sim 5 | -0.002 | 0.003 | -0.003 | -0.004 | -0.005 | -0.006 |
| sim 6 | -0.009 | -0.007 | -0.009 | -0.009 | -0.009 | -0.011 |
| sim 7 | -0.005 | 0.007 | -0.003 | -0.005 | -0.005 | -0.005 |
| sim 8 | 0.009 | 0.002 | 0.009 | 0.010 | 0.010 | 0.006 |

kable(round(sqrt(lin\_mse1),3),  
 caption='Linear Outcome Model -- Linear Regression: RMSE')

Linear Outcome Model – Linear Regression: RMSE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Baseline Logistic | Baseline Mahalanobis | CBPS - Over | CBPS - Just | EB - 1 | EB - 2 |
| sim 1 | 0.122 | 0.089 | 0.096 | 0.100 | 0.100 | 0.102 |
| sim 2 | 0.077 | 0.076 | 0.050 | 0.050 | 0.050 | 0.056 |
| sim 3 | 0.117 | 0.073 | 0.083 | 0.086 | 0.086 | 0.089 |
| sim 4 | 0.081 | 0.080 | 0.056 | 0.056 | 0.056 | 0.069 |
| sim 5 | 0.156 | 0.067 | 0.118 | 0.137 | 0.139 | 0.142 |
| sim 6 | 0.085 | 0.073 | 0.063 | 0.063 | 0.063 | 0.067 |
| sim 7 | 0.171 | 0.073 | 0.120 | 0.142 | 0.144 | 0.171 |
| sim 8 | 0.080 | 0.059 | 0.059 | 0.061 | 0.061 | 0.062 |

kable(sqrt(round(linCIcov1,3)),  
 caption='Linear Outcome Model -- Linear Regression: 95% Confidence Intervals')

Linear Outcome Model – Linear Regression: 95% Confidence Intervals

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Baseline Logistic | Baseline Mahalanobis | CBPS - Over | CBPS - Just | EB - 1 | EB - 2 |
| sim 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| sim 2 | 1 | 0 | 1 | 1 | 1 | 1 |
| sim 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| sim 4 | 1 | 0 | 0 | 0 | 0 | 0 |
| sim 5 | 1 | 0 | 1 | 1 | 1 | 0 |
| sim 6 | 0 | 0 | 0 | 0 | 0 | 0 |
| sim 7 | 1 | 0 | 1 | 0 | 0 | 1 |
| sim 8 | 0 | 1 | 0 | 0 | 0 | 0 |

apply(linCIcov1,2,mean)

## Baseline Logistic Baseline Mahalanobis CBPS - Over   
## 0.625 0.250 0.500   
## CBPS - Just EB - 1 EB - 2   
## 0.375 0.375 0.375

kable(round(lin\_bias2,3),   
 caption='Non-Linear Outcome Model -- Linear Regression: Bias')

Non-Linear Outcome Model – Linear Regression: Bias

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Baseline Logistic | Baseline Mahalanobis | CBPS - Over | CBPS - Just | EB - 1 | EB - 2 |
| sim 1 | 0.006 | 0.011 | 0.008 | 0.008 | 0.008 | 0.009 |
| sim 2 | 0.000 | 0.005 | 0.001 | 0.001 | 0.001 | 0.002 |
| sim 3 | 0.000 | -0.001 | 0.000 | 0.000 | 0.000 | 0.001 |
| sim 4 | -0.001 | -0.008 | -0.003 | -0.003 | -0.003 | -0.009 |
| sim 5 | -0.002 | 0.003 | -0.003 | -0.004 | -0.005 | -0.006 |
| sim 6 | -0.009 | -0.007 | -0.009 | -0.009 | -0.009 | -0.011 |
| sim 7 | 0.006 | 0.005 | 0.007 | 0.009 | 0.010 | 0.008 |
| sim 8 | 0.012 | 0.014 | 0.011 | 0.011 | 0.011 | 0.012 |

kable(round(sqrt(lin\_mse2),3),  
 caption='Non-Linear Outcome Model -- Linear Regression: RMSE')

Non-Linear Outcome Model – Linear Regression: RMSE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Baseline Logistic | Baseline Mahalanobis | CBPS - Over | CBPS - Just | EB - 1 | EB - 2 |
| sim 1 | 0.122 | 0.089 | 0.096 | 0.100 | 0.100 | 0.102 |
| sim 2 | 0.077 | 0.076 | 0.050 | 0.050 | 0.050 | 0.056 |
| sim 3 | 0.117 | 0.073 | 0.083 | 0.086 | 0.086 | 0.089 |
| sim 4 | 0.081 | 0.080 | 0.056 | 0.056 | 0.056 | 0.069 |
| sim 5 | 0.156 | 0.067 | 0.118 | 0.137 | 0.139 | 0.142 |
| sim 6 | 0.085 | 0.073 | 0.063 | 0.063 | 0.063 | 0.067 |
| sim 7 | 0.200 | 0.074 | 0.137 | 0.168 | 0.171 | 0.185 |
| sim 8 | 0.079 | 0.080 | 0.063 | 0.065 | 0.065 | 0.074 |

kable(sqrt(round(linCIcov2,3)),  
 caption='Linear Outcome Model -- Linear Regression: 95% Confidence Intervals')

Linear Outcome Model – Linear Regression: 95% Confidence Intervals

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Baseline Logistic | Baseline Mahalanobis | CBPS - Over | CBPS - Just | EB - 1 | EB - 2 |
| sim 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| sim 2 | 1 | 0 | 1 | 1 | 1 | 1 |
| sim 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| sim 4 | 1 | 0 | 0 | 0 | 0 | 0 |
| sim 5 | 1 | 0 | 1 | 1 | 1 | 0 |
| sim 6 | 0 | 0 | 0 | 0 | 0 | 0 |
| sim 7 | 1 | 0 | 0 | 1 | 1 | 1 |
| sim 8 | 0 | 0 | 0 | 0 | 0 | 0 |

apply(linCIcov2,2,mean)

## Baseline Logistic Baseline Mahalanobis CBPS - Over   
## 0.625 0.125 0.375   
## CBPS - Just EB - 1 EB - 2   
## 0.500 0.500 0.375

Above, the first three tables display results: (1) Bias and (2) MSE for 8 different simulations for which the true outcome model is linear. The third and fourth tables show the equivalent results, except with a non-linear true outcome model.

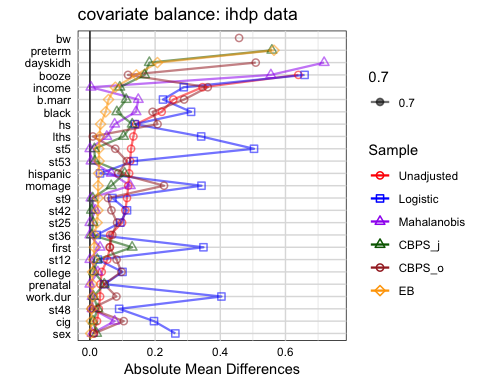
For the first set of simulations (linear true outcome model), the Mahalanobis model has the lowest RMSE for 5 out of 8 simulations. The Mahalanobis model performs particularly well when the true propensity score model is linear and generally performs worse than other models when the true propensity score model is non-linear. The CBPS and EB models generally do well, except for simulations #5 and #7. What’s odd is that simulations #5 and #7 are equivalent to #6 and #8, except that the true propensity score model is NON-linear for #6 and #8. So the intuitive expectation would be that the models would perform better and not worse when the true propensity score model is linear. The baseline logistic model exhibits this pattern as well.

## Application to IHDP Data:

In this section, I apply the matching and reweighting methods to evaluate the Infant Health and Development Program (IHDP).

The dataset for this section is from homework 4. The dataset contains personal details about approximately 4500 children born in the 1980s and their mothers. Of these 4500, 290 received the treatment: IHDP. IHDP provides special services for the children who receive treatment, such as high-quality child care in the second and third years of life. Treatment was not randomly assigned, but rather, to children who were born (1) prematurely, (2) with low birth weight (1500-2500 grams), and (3) lived in one of the eight cities where the intervention took place. The outcome of interest is a test score conducted at age 3 (similar to an IQ measure).

print(ihdp\_plot)



Simular to the results from the previous simulations, the entropy balancing model achieves the best balance on the covariates.

## Conclusion and Limitations: