E-Appendix for "Variable Selection for Propensity Score Estimation via Balancing Covariates"

1. METHODS

We first define some notation. Let Y denote the response of interest and $\mathbf{X}$ denote a p-dimensional vector of covariates. Let T denote a binary indicator of treatment exposure: $T = 1$ if treated, $T = 0$ if control. $(Y_i, \mathbf{X}_i, T_i)$, $i = 1, \ldots, n$, is a random sample from $(Y, \mathbf{X}, T)$. We further define $Y(1)$ as the potential outcome if the subject were treated and $Y(0)$ as the potential outcome if the subject were assigned to the control group.[1,2,3] Depending on the type of the outcome variable, the causal estimands can be defined as the difference between $Y(1)$ and $Y(0)$ if Y is continuous or the odds ratio if Y is binary. We further denote the propensity score as $\pi(\mathbf{X}) = \Pr(T = 1|\mathbf{X})$.[4] In observational studies, $\pi(\mathbf{X})$ is unknown and needs to be estimated from the data.

Next, we briefly review the generalized boosted model (GBM)[5] and covariate balancing propensity scores (CBPS).[6] Both approaches estimate propensity scores by achieving balance in the covariates.

1.1 Generalized Boosted Model

GBM employs a boosting algorithm to estimate propensity scores. Boosting belongs to the category of so-called ensemble methods: instead of generating one classification/regression tree, it generates multiple trees. Each tree is constructed on a bootstrapped sample of the original data. The final estimator is the aggregation of the estimates from each tree.[7] In GBM, let $g(\mathbf{X}) = \log[\pi(\mathbf{X})/(1 - \pi(\mathbf{X}))]$ and the maximum likelihood function for T given $\mathbf{X}$ can be written as:

$$l(g) = \sum_{i=1}^{n} T_i g(\mathbf{X}_i) - \log\{1 + \exp[g(\mathbf{X}_i)]\}. \tag{1}$$

To maximize $l(g)$ in equation (1), $g(\mathbf{X})$ is updated at each iteration with $g(\mathbf{X}) + \alpha h(\mathbf{X})$ where $h(\mathbf{X})$ is the fitted value from a regression tree which models $\gamma_i = T_i - 1/\{1 + \exp[-g(\mathbf{X}_i)]\}$,

the largest increase in equation (1). α is a shrinkage parameter, which is smaller than 1 to prevent the algorithm from changing too fast at each step. Since $h(\mathbf{X})$ is fitted by a tree structure, this approach is nonparametric; it picks important covariates, nonlinear terms and interaction terms by optimizing some nonparametric information criterion without specifying a parametric model.[5]

In GBM, the number of trees is an important tuning parameter. When the number of trees gets larger, the model gets more complicated and the variance of the estimates increases and the bias decreases in general. To avoid over-fitting, McCaffrey et al.[5] suggested selecting the optimal number of trees by minimizing the standardized absolute mean difference (ASAM) among the covariates. The underlying idea is that if the propensity score is correctly estimated, the covariates should be distributed almost the same among different treatment groups. Here, the distribution of the covariates is characterized by the weighted group mean of the covariates. For a single covariate x, the standardized difference is defined as:

$$d = \frac{\bar{x}^w_{treated} - \bar{x}^w_{control}}{\sqrt{(s^2_{treated} + s^2_{control})/2}},$$

where $\bar{x}^w_{treated}$ is the weighted average of x in the treatment group and $\bar{x}^w_{control}$ is the weighted average of x in the control group. When estimating the average treatment effect (ATE), $\bar{x}^w_{treated} = \frac{\sum_{i=1}^n x_i T_i/\hat{\pi}_i}{\sum_{i=1}^n T_i/\hat{\pi}_i}$ and $\bar{x}^w_{control} = \frac{\sum_{i=1}^n x_i(1-T_i)/(1-\hat{\pi}_i)}{\sum_{i=1}^n (1-T_i)/(1-\hat{\pi}_i)}$, where $\hat{\pi}_i$ is the estimated propensity score for subject i; when estimating the average treatment effect among the treated (ATT), $\bar{x}^w_{treated} = \frac{\sum_{i=1}^n x_i T_i}{\sum_{i=1}^n T_i}$ and $\bar{x}^w_{control} = \frac{\sum_{i=1}^n x_i(1-T_i)\hat{\pi}_i/(1-\hat{\pi}_i)}{\sum_{i=1}^n (1-T_i)\hat{\pi}_i/(1-\hat{\pi}_i)}$. The ASAM is simply the average of $|d|$ over all the covariates.

Although GBM automatically performs variable selection by splitting trees on different covariates, the determination of the number of trees depends on which set of covariates are included in the balancing condition, i.e., the covariates that we aim to balance. For example, in some data applications, if we include all the covariates in the GBM model and choose the optimal number of trees that leads to the smallest ASAM value, the standardized mean differences for some covariates increase after the propensity score adjustment. This is

because we focus on minimizing the average value over all the covariates. We often find that focusing on a subset of covariates may lead to better performance.

1.2 Covariate Balancing Propensity Scores

CBPS aims to estimate propensity scores by optimizing the covariate balance. Different from GBM, CBPS assumes the propensity score follows a logistic regression model, i.e.,

$$\pi_\beta(\mathbf{X}) = \frac{\exp(\mathbf{X}'\beta)}{1+\exp(\mathbf{X}'\beta)}.$$

β is solved by satisfying the balancing condition:

$$\mathrm{E}\{\frac{T\widetilde{\mathbf{X}}}{\pi_\beta(\mathbf{X})} - \frac{(1-T)\widetilde{\mathbf{X}}}{1-\pi_\beta(\mathbf{X})}\} = 0, \tag{2}$$

where $\widetilde{\mathbf{X}}$ is a function of $\mathbf{X}$ specified by the researcher. For example, $\widetilde{\mathbf{X}} = \mathbf{X}$ implies the weighted mean value of each covariate is the same between the treatment and the control group. If setting $\widetilde{\mathbf{X}} = \frac{d\pi_\beta(\mathbf{X})}{d\beta}$, we are solving the maximum likelihood estimator (MLE) of β because equation (2) is the score function for MLE.[6]

The above condition is for the estimation of ATE. For estimating ATT, the balancing condition is

$$\mathrm{E}\{T\widetilde{\mathbf{X}} - \frac{\pi_\beta(\mathbf{X})(1-T)\widetilde{\mathbf{X}}}{1-\pi_\beta(\mathbf{X})}\} = 0. \tag{3}$$

If CBPS is employed to estimate propensity scores and $\tilde{\mathbf{X}} = \mathbf{X}$, then the standardized absolute mean difference is zero for each covariate and consequently, the ASAM value equals zero. Similar to GBM, CBPS depends on the set of variables that are included in $\widetilde{\mathbf{X}}$. For simplicity, we assume $\widetilde{\mathbf{X}}$ only contains first-order terms of the selected covariates.

2. DETAILS ON THE SIMULATION STUDY

In the simulation study, we first generate three covariates, (X_1, X_2, X_3), from a standard normal distribution. Then, the treatment indicator is generated from a Bernoulli distribution with

$$\Pr(T = 1|X_1, X_2, X_3) = \Phi(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3),$$

where $(\beta_0, \beta_1, \beta_2, \beta_3) = (0, 0.5, 0, 0.75)$. The outcome variable Y is generated from a Poisson model with

$$\mathrm{E}(Y|T, X_1, X_2, X_3) = \exp\{\alpha_0 + \alpha_1((1 + \exp(-3X_1))^{-1} + \alpha_2 X_2 + \alpha_3 X_3 + \alpha_4 T\},$$

where $(\alpha_0, \alpha_1, \alpha_2, \alpha_3) = (0.5, 4, 1, 0)$ and the true treatment effect $\alpha_4 = 0.5$. Based on the simulation setup, X_1 is the real confounder which jointly affects the treatment and the outcome variable; X_3 is only related to the treatment variable while X_2 is only related to the outcome variable.

We employ two different approaches to estimate the treatment effect: inverse probability weighting (IPW) and matching. In IPW, we assign each subject a weight: for those who receive the treatment, the weight is $1/\hat{\pi}$, where $\hat{\pi}$ is the estimated propensity score from either a CBPS model or a GBM model; for those who receive the control, the weight is $1/(1 - \hat{\pi})$. Then, we conduct a weighted Poisson regression, i.e., fitting the following model: $\mathrm{E}(Y|T) = \exp\{\alpha_0 + \gamma T\}$, where $\hat{\gamma}$ is the estimated treatment effect. In matching, we perform a one-to-one matching with replacement on $\hat{\pi}$. Then, we fit an unweighted Poisson model on the matched data instead of the whole data set. Again, $\hat{\gamma}$ is the estimated treatment effect.

For CBPS, we employ the *CBPS* package in R to estimate propensity scores. For GBM, we employ the *twang* package and specify interaction.depth=1 and stop.method="es.mean", which means we only allow main effects in each fitted tree and the stopping criterion is based on ASAM. However, the *ps* function in *twang* only produces results when there are two or more covariates in the model. Therefore, we only report the results for four different models in the eTable 1.

eTable 1. Simulation results for the estimated treatment effect; the propensity scores are estimated by GBM

	X_1, X_2	X_1, X_3	X_2, X_3	X_1, X_2, X_3	Reference
$n = 500$					
			Inverse Probability Weighting		
Bias	0.0832	0.1311	0.7313	0.1565	0.0080
Var	0.0240	0.0489	0.0356	0.0338	0.0289
MSE	0.0309	0.0661	0.5704	0.0582	0.0290
			Matching		
Bias	-0.0757	-0.0314	0.7614	-0.0410	0.0043
Var	0.0193	0.0676	0.0427	0.0620	0.0362
MSE	0.0250	0.0685	0.6223	0.0637	0.0362
$n = 2500$					
			Inverse Probability Weighting		
Bias	0.0361	0.0671	0.7384	0.0833	-0.0029
Var	0.0046	0.0121	0.0067	0.0080	0.0061
MSE	0.0059	0.0166	0.5519	0.0150	0.0061
			Matching		
Bias	-0.0344	-0.0240	0.7480	-0.0224	0.0002
Var	0.0029	0.0147	0.0078	0.0105	0.0064
MSE	0.0040	0.0153	0.5673	0.0110	0.0064

Lastly, in the simulation study, we only allow main effects for each fitted tree by GBM. Similarly, for CBPS, we let $\widetilde{\mathbf{X}} = \mathbf{X}$. However, for both algorithms, more flexible models can be fitted. For example, in GBM, we may specify the model to include at most three-way interactions in the covariates. For CBPS, we can add $(X_1^2, \ldots, X_p^2)$ into the balancing conditions in equation (2) or equation (3). Future studies may investigate how robust our findings are to more complex models.

3. R CODES

The following is the R codes for the simulation study:

```
library(CBPS)
library(twang)
library(Matching)

n = 500
repe =1000
fnames=paste("sim",1:repe,".dat",sep="")

for (i in 1:repe){
    X1 = rnorm(n,0,1)
    X2 = rnorm(n,0,1)
    X3 = rnorm(n,0,1)
    A = rbinom(n,1,pnorm(0 + 0.5*X1 + 0*X2 + 0.75*X3,0,1))
    Y = rpois(n, exp(0.5 + 4*(1/(1+exp(-3*X1))-0.5)+1*X2+0*X3 + 0.5*A))
    write.table(cbind(A,Y,X1,X2,X3),file=fnames[i],row.names=FALSE,
    col.names=FALSE,append=FALSE)
}

gamma = matrix (NA,repe,12)
gamma2= matrix (NA,repe,12)
wt=matrix(NA,n,12)
PS = matrix(NA,n,12)

for (i in 1:repe){
    data=read.table(file=fnames[i],header=FALSE)
    A=data[,1]
    Y=data[,2]
    X1=data[,3]
    X2=data[,4]
    X3=data[,5]

    PS[,1]=CBPS(A ~ X1, ATT = FALSE, method = "exact")$fitted.values
    PS[,2]=CBPS(A ~ X2, ATT = FALSE, method = "exact")$fitted.values
    PS[,3]=CBPS(A ~ X3, ATT = FALSE, method = "exact")$fitted.values
    PS[,4]=CBPS(A ~ X1+X2, ATT = FALSE, method = "exact")$fitted.values
    PS[,5]=CBPS(A ~ X1+X3, ATT = FALSE, method = "exact")$fitted.values
    PS[,6]=CBPS(A ~ X2+X3, ATT = FALSE, method = "exact")$fitted.values
    PS[,7]=CBPS(A ~ X1+X2+X3, ATT = FALSE, method = "exact")$fitted.values

    dataset=data.frame(A=A,X1=X1,X2=X2,X3=X3)
    model1 <- ps(A ~ X1+X2, data = dataset,interaction.depth=1,
```

```
    stop.method = "es.mean")
    PS[,8] <- as.matrix(model1$ps)[,1]
    model2 <- ps(A ~ X1+X3, data = dataset, interaction.depth=1,
    stop.method = "es.mean")
    PS[,9] <- as.matrix(model2$ps)[,1]
    model3 <- ps(A ~ X2+X3, data = dataset, interaction.depth=1,
    stop.method = "es.mean")
    PS[,10] <- as.matrix(model3$ps)[,1]
    model4 <- ps(A ~ X1+X2+X3, data = dataset, interaction.depth=1,
    stop.method = "es.mean")
    PS[,11] <- as.matrix(model4$ps)[,1]

    PS[,12] = glm(A ~ X1+X2, family = "binomial" (link = "probit"))$fitted.values

 # Inverse Probability Weighting
    for (j in 1:12){
         wt[,j]=1/PS[,j]*A+1/(1-PS[,j])*(1-A)
    }
    for (j in 1:12){
    gamma[i,j] = glm(Y ~ A, weights=wt[,j], family = "poisson")$coef[2]
    }

 # Matching
    mydata=data.frame(A=A,Y=Y)
    for (j in 1:12){
        mydata.treat = mydata[Match(Y=NULL, Tr=A, X=PS[,j],
        estimand="ATE")$index.treated,]
        mydata.control = mydata[Match(Y=NULL, Tr=A, X=PS[,j],
        estimand="ATE")$index.control,]
        wholedata=rbind(mydata.treat,mydata.control)
        gamma2[i,j]=glm(Y~A,family="poisson",data=wholedata)$coef[2]
    }

}

bias = apply(gamma, FUN = mean, MARGIN = 2) - 0.5
variance = apply(gamma, FUN = var, MARGIN = 2)
MSE = apply((gamma - 0.5)^2, FUN = mean, MARGIN = 2)

bias2 = apply(gamma2, FUN = mean, MARGIN = 2) - 0.5
variance2 = apply(gamma2, FUN = var, MARGIN = 2)
MSE2 = apply((gamma2 - 0.5)^2, FUN = mean, MARGIN = 2)
```

References

1. Rubin DB. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*. 1974; 66(5): 688-701.

2. Holland P. Statistics and causal inference (with discussion). *Journal of the American Statistical Association*. 1986; 81(396), 945 – 970.

3. Lunceford JK and Davidian M. Stratification and weighting via the propensity score in estimation of causal treatment effects: a comparative study. *Statistics in Medicine*. 2004; 23(19), 2937 – 2960.

4. Rosenbaum PR and Rubin DB. The central role of the propensity score in observational studies for causal effects. *Biometrika*. 1983; 70(1), 41 – 55.

5. McCaffrey DF, Ridgeway G and Morral AR. Propensity score estimation with boosted regression for evaluating causal effects in observational studies. *Psychological Methods*. 2004; 9(4), 403–425.

6. Imai K and Ratkovic M. Covariate balancing propensity score. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. 2014; 76(1), 243–263.

7. Freund Y and Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*. 1997; 55, 119–139.