Guidelines for Using State-of-the-Art Methods to Estimate Propensity Score and Inverse Probability of Treatment Weights When Drawing Causal Inferences

Session 4. Alternative Propensity Score Estimation Methods

Beth Ann Griffin
Dan McCaffrey

Propensity Score Estimation

- We have discussed two methods for propensity score estimation
  - GBM
  - Logistic regression
- Many other methods exist
  - Super learning (Polley and van der Laan)
  - High Dimensional Propensity Scores (hd-PS, Schneeweiss and Rassan)
  - Covariate Balancing Propensity Scores (CBPS, Imai and Ratkovic)
  - Entropy Balancing, exponential tilting, minimum discriminant information adjustment (Hainmueller, Graham et al., Haberman)

Achieving Balance in the Parameter Estimation Criteria

- For GBM and logistic regression we use balance to guide model selection
  - Art of logistic regression is picking and choosing terms to get good balance and improving balance guides choice – similar to using AIC or BIC in other contexts
  - For GBM we use balance to pick the number of iterations which controls the variables included and functional form of the models – replaces cross-validation
- Parameter estimation criteria does not involve balance
  - For logistic regression, we find the MLE for the parameters of the model at each step of modeling process
  - For GBM, the algorithm select the tree models that maximize the likelihood for each iteration
- What if it did?

Achieving Balance in the Parameter Estimation Criteria

- **For logistic regression**
  $p(X, \beta) = Pr(Z = 1|X) = 1/(1 + \exp(-\beta' X))$

- $\ell(X; \beta)$ **is the log likelihood and find the value of** β **to maximize** $\ell(X; \beta)$ **or find** β **that solves** $\psi(X; \beta) = 0$

- **Suppose we added the criteria that** β **should also satisfy** $\sum_{i=1}^{n} \frac{1}{p(X,\beta)} X_i Z_i = \sum_{i=1}^{n} \frac{1}{1-p(X,\beta)} X_i (1 - Z_i)$ **when estimating the coefficients**

- **We are no longer just using balance to pick the** X**s (including interactions and polynomial terms)**

- **Using balance to determine the values of** β **– coefficient values will differ from traditional logistic regression fit**

Covariate Balancing Propensity Scores

- Covariate Balancing Propensity Scores (CBPS) follows this logic
- For logistic regression, maximizing the likelihood solves the estimating equation $\psi(X;\beta) = 0$
- CBPS includes balance in the parameter estimation criteria when estimating the parameters of the logistic regession model
  - Extends estimating equation from $\psi(X;\beta) = 0$ to

$$\left(\begin{array}{c} \psi(X;\beta) \\ \sum_{i=1}^{n} \frac{1}{p(X,\beta)} X_i Z_i - \sum_{i=1}^{n} \frac{1}{1-p(X,\beta)} X_i (1 - Z_i) \end{array} \right) = 0$$

- Developed by Imai and Ratkovic (2014) who also developed the R package CBPS to implement the method

SuperLearner Ensemble (using WeightIt)

- Greifer developed the WeightIt package to implement SuperLearner, among other methods, with ease.
- The “super” method in WeightIt estimates propensity scores using the SuperLearner algorithm (Polley et al.) for stacking predictions and then converting those propensity scores into weights
- The selected ensemble estimates propensity scores as the predicted probability of being in each treatment given covariates
- We use all possible methods for the ensemble, including gbm, glm, glmnet, randomForest, xgboost among others

Selecting Weights to Get Exact Balance

- **For a given reasonable target, τ_1 or τ_0, there are weights such that $\sum_{i=1}^{n} w_i X_i Z_i = \tau_1$ and $\sum_{i=1}^{n} w_i X_i (1 - Z_i) = \tau_0$**
- **For example, in ATT, we would let $\tau_0 = \bar{X}_1$, the vector of treatment group means, and find weights that give exact balance**
  - **Such weights will exists as long as groups are not too distinct**
  - **Multiple sets of such weights exist!**

Entropy Balancing or MDIA Weights

- For entropy balance or minimum discriminant information adjustment (MDIA), select weights to
  1. Minimize $\sum_{(i|Z=0)} w_i log(w_i)$
  2. Subject to

     $\sum_{(i|Z=0)} w_i = 1$

     $\sum_{(i|Z=0)} w_i X_i = \tau_0$
- $\sum_{(i|Z=0)} w_i log(w_i)$ is the Kullback discriminant information or Kullback-Leibler distance comparing weights to equal weighting

Form for Entropy Balance Weights

- Entropy balancing weights are of the form $\exp(\alpha + \gamma' X)$
- Same form as ATT weights ($p(X)/[1 - p(X)]$) with logistic regression propensity scores using X
  - But the coefficients will differ

Alternative Criteria for Exact Balance Weights

- Can replace the Kullback-Leibler distance with other distance measure
- Zubizarreta (2015) minimized variance of the weights
- Survey sampling calls these Generalized Regression weighting and consider several alternative distances (see Deville and Särndal, 1992 for examples)
- Entropy balancing has nice property that weights are positive

Minimal Approximately Balancing Weights

- $|\sum_{(i|Z=0)} w_i X_{ki} - \tau_0| < \delta_k$ for $k = 1, \ldots, K$
- **Select δ_k to control variance of the weights and MSE of the treatment effect estimator**

Selecting X

- Entropy balancing weights give exact balance to linear function of covariates used in balancing $\beta' X$
- If $E[Y_0 \mid X]$ is not linear in X then there can be remaining bias
- We can balance functions of the covariates to create "X" for balancing
  - For example, include covariates and their squares and cross-products

Exact Balance Versus Propensity Scores

- If we can obtain exact balance why bother with propensity scores?
  - Exact balance only on selected Xs and only for the means
  - Not clear how well other functions of covariates will balance
  - With good propensity score model all functions of covariates will balance (at least in expectation) if strong ignoribility holds
  - We have model building schemes so propensity score models will tend to be good (for large samples)
  - We don't have modeling building schemes for picking the covariates and functions of the covariates to balance exactly
  - Exact balance may come at the price of greater variability in the weights (smaller effective sample sizes) – that may not be useful if we pick the wrong covariates or functions of covariates to exactly balance

Using Exact Balance

- Pick functions of the covariates and obtain exact balance
- Obtain propensity scores and the apply exact balance to selected covariates on top of propensity score weighting
- `ebal` package in R and `ebalance` package in Stata will conduct entropy balancing

Example: CBPS with the AOD Data

- Compare the Usual care and MET/CBT-5 conditions from AOD data
- Test the relative effects of two treatment among youth like those that receive usual care
- Treatment on the treated with Usual Care as “the treated” and MET/CBT-5 as the “control”
- For this demonstration we use casewise deletion to remove records with incomplete covariate data
- Conduct the analysis in R

Prepare the Data

```
library(CBPS)

aod <- read.csv("AOD.csv")
atmeat <- subset(aod,
                 subset=(trtvar %in% c("ATM", "EAT")))
nrow(atmeat)

atmeat <- na.omit(atmeat)
nrow(atmeat)

atmeat$race4g <- as.factor(atmeat$race4g)
```


Compare to Logistic Regression: Fit the Model

- CBPS is a modification of standard logistic regression model for propensity scores, so we will compare the results of logistic regression to CBPS
- Use the covariates discussed earlier

```
plog <- glm(atm ~ age + female + race4g + sfs + sps +
                  sds + ias + ces + eps + imds + bcs +
                  prmhtx, family=binomial, data=atmeat)

atmeat$ps1 <- ifelse(atmeat$atm==1, 1,
                              exp(predict(plog)))
```


Compare to Logistic Regression: Use dx.wts to Check Balance

```
b1 <- dx.wts(atmeat$ps1,
             data=atmeat,
             vars=c("age", "female", "race4g", "sfs",
                    "sps", "sds", "ias", "ces", "eps",
                    "imds", "bcs", "prmhtx"),
             treat.var="atm",
             estimand="ATT",
             x.as.weights=TRUE,
             sampw=NULL,
             perm.test.iters=0)
```


Run CBPS: Fit the Model and Get Weights

```
pcbps <- CBPS(atm ~ age + female + race4g + sfs +
                    sps + sds + ias + ces + eps +
                    imds + bcs + prmhtx,
               data=atmeat, ATT=TRUE)

atmeat$ps2 <- pcbps$weights
```


Run CBPS:
Use dx.wts to Check Balance

```
b2 <- dx.wts(atmeat$ps2,
             data=atmeat,
             vars=c("age", "female", "race4g", "sfs",
                    "sps", "sds", "ias", "ces", "eps",
                    "imds", "bcs", "prmhtx"),
             treat.var="atm",
             estimand="ATT",
             x.as.weights=TRUE,
             sampw=NULL,
             perm.test.iters=0)
```


Run GBM: Fit the Model and Get Weights

```
pgbm <- ps(atm ~ age + female + race4g + sfs +
                 sps + sds + ias + ces + eps +
                 imds + bcs + prmhtx,
           data=atmeat, estimand="ATT", n.trees=10000,
           stop.method="es.max")

atmeat$ps3 <- unlist(pgbm$w)
```


Run SuperLearner:
Fit the Model and Get Weights

```
SL.library = listWrappers(what="SL")
SL.library = SL.library[startsWith(SL.library, "SL.")]
SL.library = SL.library[!(SL.library %in%
      c("SL.bartMachine", "SL.svm", "SL.template"))]

pSL <- weightit(atm ~ age + female + race4g + sfs + sps +
                  sds + ias + ces + eps + imds + bcs +
                  prmhtx, data = atmeat, method = "super",
                estimand = "ATT", SL.library = SL.library)

atmeat$psSL = pSL$weights
```


Run SuperLearner:
Use dx.wts to Check Balance

```
bSL <- dx.wts(atmeat$psSL,
             data=atmeat,
             vars=c("age", "female", "race4g", "sfs",
                    "sps", "sds", "ias", "ces", "eps",
                    "imds", "bcs", "prmhtx"),
             treat.var="atm",
             estimand="ATT",
             x.as.weights=TRUE,
             sampw=NULL,
             perm.test.iters=0)
```


Compare Weights

- Compare the summary tables to check overall balance and effective sample sizes

```
b1$summary
b2$summary
summary(pgbm)
bSL$summary
```

Compare Weights

- Compare the summary tables to check overall balance and effective sample sizes

```
rbind(b1$summary[2,],b2$summary[2,],summary(pgbm)[2,-8],bSL$summary[2,])
  type       n.treat n.ctrl ess.treat ess.ctrl    max.es   mean.es
1                442   2408       442  84.3727 0.6248657 0.1277555
2                442   2408       442 254.6165 0.0096218 0.0052786
3 es.max.ATT     442   2408       442 498.6925 0.2944589 0.0892948
4                442   2408       442 442.5542 0.3514092 0.0971101
     max.ks   mean.ks iter
1 0.2120477 0.0782432   NA
2 0.2579716 0.0545864   NA
3 0.1491926 0.0535904  710
4 0.1714221 0.0602218   NA
```

- Substance use frequency and illegal activities remain imbalanced with logistic weights

- The environment scale remains imbalanced with GBM (only ES $>$ 0.20)

Compare ATT Estimates

```
d1 <- svydesign(id=~1, weights=~ps1, data=atmeat)
d2 <- svydesign(id=~1, weights=~ps2, data=atmeat)
d3 <- svydesign(id=~1, weights=~ps3, data=atmeat)
dSL <- svydesign(id=~1, weights=~psSL, data=atmeat)

f1 <- svyglm(sfs8p12 ~ atm, design=d1)
f2 <- svyglm(sfs8p12 ~ atm, design=d2)
f3 <- svyglm(sfs8p12 ~ atm, design=d3)
f4 <- svyglm(sfs8p12 ~ atm, design=dSL)

res <- list(logit=summary(f1)$coef,
            cbps=summary(f2)$coef,
            gbm=summary(f3)$coef,
            SL=summary(f4)$coef)
```


Compare ATT Estimates

```
> print(res)
$logit
              Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 0.10139089 0.01884309 5.3807991 8.017795e-08
atm         0.01297383 0.01999528 0.6488445 5.164912e-01

$cbps
              Estimate  Std. Error  t value     Pr(>|t|)
(Intercept) 0.08779229 0.009401344 9.338270 1.912081e-20
atm         0.02657243 0.011538391 2.302958 2.135292e-02

$gbm
              Estimate  Std. Error   t value     Pr(>|t|)
(Intercept) 0.09421772 0.006259164 15.052766 2.487814e-49
atm         0.02014700 0.009161131  2.199183 2.794505e-02

$SL
              Estimate  Std. Error   t value     Pr(>|t|)
(Intercept) 0.10011597 0.006874258 14.563894 2.129116e-46
atm         0.01424875 0.009591903  1.485498 1.375227e-01
```


Get Exact Balance Using Entropy Balancing or MDIA Weights

- **The package `ebal` and the function `ebalance` find the entropy balancing or MDIA weights**

- **`ebalance` requires a data matrix of covariates including transforming class variables, I use `model.matrix` to generate this matrix**

```
library(ebal)

tmp <- model.matrix(atm ~ age + female + race4g + sfs +
                         sps + sds + ias + ces + eps +
                         imds + bcs + prmhtx, data=atmeat)

  ## drop the intercept ##
tmp <- tmp[,-1]
```

Get Exact Balance Using Entropy Balancing or MDIA Weights: Run Fit and Get Weights

```
pbal <- ebalance(Treatment=atmeat$atm, X=tmp)

atmeat$ps4 <- 1
atmeat$ps4[atmeat$atm==0] <- pbal$w
```

- Generates weights only for the control cases

Compare Weights

- Compare the summary tables to check overall balance and effective sample sizes

```
rbind(b1$summary[2,],b2$summary[2,],
      summary(pgbm)[2,-8], bSL$summary[2,], b4$summary[2])
  type n.treat n.ctrl ess.treat  ess.ctrl     max.es      mean.es
2           442    2408       442   84.3727  0.6248657    0.1277555
2           442    2408       442  254.6165  0.0096218   0.00527861
es.max.ATT 442    2408       442  498.6925  0.2944589    0.0892948
2           442    2408       442  442.5542  0.3514092   0.09711015
2           442    2408       442  257.4996  2.21122e-05  4.91149e-06
             max.ks    mean.ks    iter
2            0.2120477 0.0782432    NA
2            0.2579716 0.05458636   NA
es.max.ATT   0.1491926 0.05359044  710
2            0.1714221 0.06022182   NA
2            0.2613977 0.05403091   NA
```

- `ebalance` yields ES of effectly zero but the KS is not zero, only balances the means

Compare ATT Estimates

```
logit
              Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 0.10139089 0.01884309 5.3807991 8.017795e-08
atm         0.01297383 0.01999528 0.6488445 5.164912e-01

cbps
              Estimate  Std. Error  t value     Pr(>|t|)
(Intercept) 0.08779229 0.009401344 9.338270 1.912081e-20
atm         0.02657243 0.011538391 2.302958 2.135292e-02

gbm
              Estimate  Std. Error   t value     Pr(>|t|)
(Intercept) 0.09421772 0.006259164 15.052766 2.487814e-49
atm         0.02014700 0.009161131  2.199183 2.794505e-02

SL
              Estimate  Std. Error   t value     Pr(>|t|)
(Intercept) 0.10011597 0.006874258 14.563894 2.129116e-46
atm         0.01424875 0.009591903  1.485498 1.375227e-01

ebal
              Estimate  Std. Error  t value     Pr(>|t|)
(Intercept) 0.08734734 0.009234442 9.458866 6.291983e-21
atm         0.02701738 0.011402811 2.369361 1.788516e-02
```


Compare Coefficients

	logit	CBPS	entropy
(Intercept)	-3.24975893	-2.13708597	-2.189729100
age	0.03101638	-0.03416526	-0.029536458
female	-0.17825380	-0.30058025	-0.322173568
race4g2	0.33919320	0.32815272	0.348741470
race4g3	-1.14100021	-1.05909266	-1.061779956
race4g4	-0.93096936	-0.96710013	-0.985629576
sfs	-0.05263512	-0.26199278	-0.258247705
sps	0.03433795	0.07466073	0.070599036
sds	0.04362259	-0.01342978	-0.008163907
ias	6.23940874	4.98807125	4.920726580
ces	2.71148703	2.77443999	2.801470226
eps	3.49192440	3.37508938	3.261896874
imds	-0.09699256	-0.08302634	-0.080216288
bcs	-0.02381459	-0.02660521	-0.023874070
prmhtx	-0.02625957	-0.16914946	-0.199614380

Combining GBM and Entropy

- GBM gets good balance but not exact mean balance
- We could combine GBM with entropy to get exact balance on the means of all or select variables while still fitting a more flexible propensity score model

Combining GBM and Entropy (2)

- `ebalance` with ias as the only covariate because GBM did not balance the mean for this variable
- Improved mean balance: max.es fell from 0.29 for GBM alone to 0.16 with GBM and entropy
- Reduced ESS from 498 to 248
- Improved balance comes at the cost of more variable weights and possibly less precision in ATT estimate

Combining GBM and Entropy (3)

- `ebalance` using all covariates
- Improved mean balance: max.es fell from 0.29 for GBM alone to zero with GBM and entropy
- Reduced ESS from 498 to 158
- Further improvement really reduces the ESS

Combining GBM and Entropy – R Code

```
pbal2 <- ebalance(Treatment=atmeat$atm,
                  X=as.matrix(tmp[,"ces"]),
                  get.weights(pgbm)[atmeat$atm==0])

pbal4 <- ebalance(Treatment=atmeat$atm,
                  X=tmp,
                  get.weights(pgbm)[atmeat$atm==0])
```


SAS Code for Running CBPS

```
%CBPS(treatvar=atm,
          vars=age female race4g sfs sps sds
               ias ces eps imds bcs prmhtx,
          class=race4g,
          dataset=atmeat,
          estimand=ATT,
          method=over,
          output_dataset=cbpswts,
          permtestiters=0,
          Rcmd=C:\Program Files\R\R-3.0.2\bin\x64\r.EXE,
          objpath=c:\Users\dmccaffrey\twang);
```

- Returns dataset with the weights and prints summary and balance table information