Data Adaptive Estimation of the Treatment Specific Mean in Causal Inference

R-package `cvDSA`

Yue Wang

Division of Biostatistics

Nov. 2004

Outlines

- Introduction: Data structure and Marginal Structural Model.
- Estimation Road map
  - Choice of loss function;
  - Generating candidate estimators;
  - Selection among candidate estimators: cross-validation;
  - D/S/A algorithm for computing the optimal index set;
  - Selection of nuisance parameter models.
- R-package `cvDSA`
  - Data-adaptive estimation for nuisance parameter model (`cvGLM()`);
  - Data-adaptive estimation for the Marginal structural model (`cvMSM()`).

Data structure and Marginal Structural Model

- Full data structure.

  $X = ((Y_a, a \in \mathcal{A}), W) \sim F_{X,0}$

  Y_a is the counterfactual outcome, a represents treatment, W represents the baseline covariates.

- Observed data structure.

  $O = (A, Y_A, W) \sim P_0 = P_{F_{X,0}, g_0}$

  A is a random variable denoting which treatment is assigned, Y_A is the outcome under treatment A.

- Marginal Structural Model (MSM).

  Estimate treatment specific mean $E(Y_a|V)$ as a function of a and V, where $V \subset W$.

  Randomization assumption (RA): treatment is randomly assigned within strata of W, $g_0(a|X) = g_0(a|W)$ for all $a \in \mathcal{A}$.

- Defining the parameter of interest in terms of a loss function.

  Let $\psi(a, v) = E(Y_a|V)$ be the parameter of interest. The true parameter value ψ_0 is the one maps the true data population, $\psi_0 \equiv \psi(F_{X,0})$. It is defined in terms of a loss function, $L(X, \psi)$, as the minimizer of the expected loss, or risk. That is, ψ_0 is

$$\psi_0 = \arg\min_{\psi \in \Psi} E(L(X, \psi))$$

- Full data loss function.

$$L(X, \psi) = \sum_{a \in \mathcal{A}} (Y_a - \psi(a, v))^2$$

  The true model ψ_0 is the minimizer of the expectation of the loss function.

Estimation Road Map: Choices of loss function

- Choices of mapping the full data loss function

  The three mappings of the the full data loss function have the same expectation as the full data loss function.

  1. G-computational mapping

$$
\begin{aligned}
L_{Gcomp}(O, \psi|\eta_0) &= IC(O|Q_0, L(X, \psi)) \\
&= \sum_{a \in \mathcal{A}} E((Y - \psi(A, V))^2 | A = a, W) \\
&= \sum_{a \in \mathcal{A}} \{E(Y^2 | A = a, W) \\
&\quad -2E(Y|A = a, W)\psi(a, v) \\
&\quad +\psi(a, v)^2\}
\end{aligned}
$$

2. IPTW mapping

$$\begin{aligned} L_{IPTW}(O,\psi|\eta_0) &= IC(O|g_0, L(X,\psi)) \\ &\equiv \frac{(Y-\psi(A,V))^2}{g(A|X)} g(A|V); \end{aligned}$$

3. Double Robust mapping (by van der Laan and Robins (2002))

$$\begin{aligned} L_{DR}(O,\psi|\eta_0) &= IC(O|Q_0, g_0, L(\cdot,\psi)) \\ &= \frac{(Y-\psi(A,V))^2}{g(A|X)} g(A|V) \\ &\quad -\frac{g(A|V)}{g(A|X)} E\left[(Y-\psi(A,V))^2|A,W\right] \\ &\quad + \sum_{a\in\mathcal{A}} E\left[(Y-\psi(A,V))^2|A=a,W\right] g(a|V), \end{aligned}$$

Estimation Road Map: Generating candidate estimators

- The minimum empirical risk estimator

$$\text{argmin}_{\psi\in\mathbf{\Psi}} \int L(o,\psi \mid v_n) dP_n(o)$$

typically suffers from the curse of dimensionality due to the size of Ψ. A general approach is to construct a sequence or collection of subspaces approximating the whole parameter space $\mathbf{\Psi}$, a so called **sieve** , and select the actual subspace whose corresponding minimum empirical risk estimator minimizes an appropriately penalized empirical risk or a cross-validated empirical risk.

- Let $\{\Psi_k\}$ be a sieve and $\Psi_k \subset \Psi$, define

$$\mathbf{\Psi} = \left\{ g\left(\sum_{j\in I} \beta_j \phi_j\right) : I \subset \mathcal{I}, \beta \right\},$$

where ϕ_j is a tensor product of basis functions. Choose univariate function $e_k(W) = W^k$ as the basis function, I is a vector which represents for a polynomial.

Given a vector $\vec{p} = (p_1, \ldots, p_d) \in \mathbb{N}^d$, the tensor product identified by $\vec{p}$ is:

$$\begin{aligned} \phi_{\vec{p}} &= e_{p_1}(W_1) \times \ldots \times e_{p_d}(W_d) \\ &= W_1^{p_1} \ldots W_d^{p_d}. \end{aligned}$$

- Define a collection of subspaces as $\mathbf{\Psi}_s \subset \mathbf{\Psi}$, indexed by an s. Such subspaces are obtained by restricting the subsets I of basis functions to be contained in $\mathcal{I}_s \subset \mathcal{I}$, and/or restricting the values for the corresponding coefficients $(\beta_{\vec{p}} : \vec{p} \in I)$ to be contained in $B_{I,s} \subset B_I$:

$$\mathbf{\Psi}_s = \{\psi_{I,\beta} : I \in \mathcal{I}_s \subset \mathcal{I}, \beta \in B_{I,s} \subset B_I\}.$$

- For each s, compute (or approximate as best as one can) the minimizer of the empirical risk over the subspace $\boldsymbol{\Psi}_s$:

$$\hat{\Psi}_s(P_n) \equiv \text{argmin}_{\psi \in \Psi_s} \int L(o, \psi \mid \upsilon_n) dP_n(o).$$

  - Step 1. Given each possible subset $I \in \mathcal{I}_s$ of basis functions, compute the corresponding minimum risk estimator of β:

$$\beta(P_n \mid I, s) \equiv \text{argmin}_{\beta \in B_{I,s}} \int L\left(o, \psi_{I,\beta} \mid \upsilon_n\right) dP_n(o);$$

    For each I, this results in an estimator $\psi_{I,s,n} = \hat{\Psi}_{I,s}(P_n) \equiv \psi_{I,\beta(P_n \mid I,s)}$.
  - Step 2. Minimize the empirical risk over all allowed subsets $I \in \mathcal{I}_s$ of basis functions. Specifically, one needs to minimize the function $f_E : \mathcal{I}_s \rightarrow \mathbb{R}$ defined by

$$f_E(I) \equiv \int L\left(o, \widehat{\Psi}_{I,s}(P_n)\right) dP_n(o).$$

Estimation Road Map: Selection among candidate estimators: cross-validation

- Select s with cross-validation

Cross-validation : the observations in the training set (P^0) are used to estimate the parameters and the observations in the validation set (P^1) are used to access performance of the estimators. The cross-validation selector is the chosen to have the best performance on the validation sets.

Given an estimator $\hat{\Upsilon}$ of the nuisance parameter υ_0, the cross-validation selector of s is now defined as follows:

$$\hat{S}(P_n) \equiv \text{argmin}_s E_{B_n} \int L(o, \hat{\Psi}_s(P^0_{n,B_n}) \mid \hat{\Upsilon}(P^0_{n,B_n}))dP^1_{n,B_n}(o).$$

Estimation Road Map: D/S/A algorithm for computing the optimal index set

- The goal is to estimate

$$I_0(P_n) \equiv \arg\min_{I \in \mathcal{I}} \int L(o, \hat{\Psi}_I(P_n) \mid \upsilon_0) dP_0(o).$$

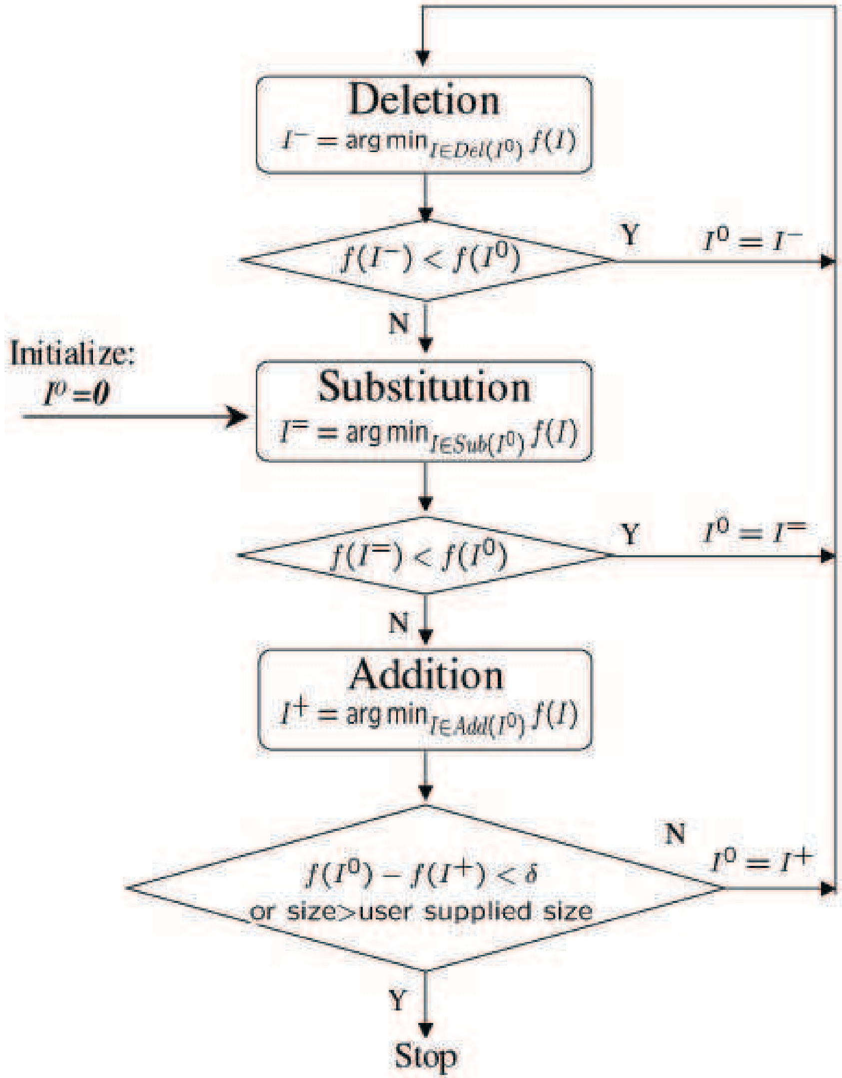
Estimation of $I_0(P_n)$ involves a two-stage procedure:

- Find the best choice within $\mathcal{I}_s$ using the empirical risk function, to find the best choice within $\mathcal{I}_s$;
- Find the best choice of s using the cross-validated risk function.

The D/S/A algorithm (Sinisi and van der Laan (2004)) maps the current index set $I^0 \in \mathcal{I}$ of size k into three collections of index sets, namely, deletion set $DEL(I^0)$, substitution set $SUB(I^0)$, and addition set $ADD(I^0)$, of size $k-1$, k and $k+1$, respectively. Let $I^0 = \{\vec{p}_1^0, \ldots \vec{p}_k^0\}$ denote the current index set, where $\vec{p}_i^0 \in \mathbb{N}^d$, $i = 1, 2, \cdots, k$:

- $DEL(I^0)$ is a set of index sets I where the i^{th} vector $\vec{p}_i^0$ is deleted from I^0, for $i = 1, 2, \cdots, k$;
- $SUB(I^0)$ is a set of index sets I where the i^{th} vector $\vec{p}_i^0$ is substituted by one of the new vectors $\vec{p}_{ij} = \vec{p}_i^0 + \delta e_j$, where $\delta = \{-1, 1\}$, $j = 1, 2, \cdots, d$, for $i = 1, 2, \cdots, k$;
- $ADD(I^0)$ is a set of index sets I obtained by adding one of the unit vector e_j or one of the new vectors $\vec{p}_{ij}$ in $SUB(I^0)$ to I^0, $j = 1, 2, \cdots, d$, for $i = 1, 2, \cdots, k$.

Deletion/Substitution/Addition Algorithm


Deletion
$I^- = \arg\min_{I \in Del(I^0)} f(I)$
$f(I^-) < f(I^0)$
Y
$I^0 = I^-$
N
Initialize:
$I^0 = 0$
Substitution
$I^= = \arg\min_{I \in Sub(I^0)} f(I)$
$f(I^=) < f(I^0)$
Y
$I^0 = I^=$
N
Addition
$I^+ = \arg\min_{I \in Add(I^0)} f(I)$
$f(I^0) - f(I^+) < \delta$
or size>user supplied size
N
$I^0 = I^+$
Y
Stop

Estimation Road Map: Selection of nuisance parameter models

- Selecting the nuisance parameter models with CV/DSA algorithm

$$\upsilon = \{g(A|V), g(A|W), Q(Y|A,W), Q(Y^2|A,W)\}$$

Since these nuisance parameters are either observed data densities or regressions, we can estimate them with the loss-based estimation approach based on either the squared error loss function, or the minus log loss function.

R-package cvDSA

- `cvGLM()`: Selecting/Fitting Linear Models;
- `cvMSM()`: Selecting/Fitting Marginal Structural Models;
- `create.obs.data()`: Generating an observed data set;
- `check.ETA()`: Checking ETA Assumption for MSM.

R-package cvDSA

- Example 1. Generating an observed data set.

  Let sample size $N = 2000$, $W = \{W_1, W_2\}$, $W_1 \sim U(0, 1)$, $W_2 \sim U(0, 1)$, the treatment model is

$$g(A|W) = logit^{-1}(1 - W_1 + W_2),$$

  the F_x-part model is

$$Q(Y|A, W) = 1 + 2A + 1.5W_1 + W_2 - W_1 \times W_2.$$

Code:

```
n <- 1000
w1 <- runif(n, 0, 1);
w2 <- runif(n, 0, 1);
w <- cbind(w1=w1, w2=w2);

model.aw <- list(formula=list(c(1,0),c(0,1)),
coef=c(1,-1,1));
model.yaw <- list(formula=list(c(1,0,0),c(0,1,0),
c(0,0,1), c(0,1,1)), coef=c(1, 2, 1.5, 1, -1));

obs.data <- create.obs.data(w, afamily='binomial',
yfamily='gaussian', model.yaw, model.aw)
```

R-package cvDSA

- Example 2. selecting the nuisance parameter models.
  Code:

```
a<-obs.data$a
cv.model.aw<-cvGLM(y=a, x=w, ncv=5, yx.model=list(Size=3,
Order=c(2,2), Int=2), myfamily='binomial',
printout=T, detail=T)

y<-obs.data$y
cv.model.yaw<-cvGLM(y=y, x=cbind(a,w), ncv=5,
yx.model=list(Size=5, Order=c(1,2,1), Int=2),
printout=T)
```

Result:

$g(A|W)$:

```
CV selects: size =  2 , interactions =  2

with min.risk: 0.5584379
$Formula [1] "Intercept +  w1 + w2"

$Coefficients
(Intercept)          w1          w2
   1.204914   -1.356494    1.080563
```

$E(Y|A,W)$:

```
CV selects: size =  4 , interactions =  2
with min.risk: 1.018344

$Formula [1] "Intercept +  a + w1 + w2 + w1*w2"

$Coefficients

(Intercept)           a          w1          w2       w1*w2
   0.959001    2.002093    1.475317    1.089650   -1.026595
```

R-package cvDSA

- Example 3. selecting the marginal structural model.

  Code:

```
a<-obs.data$a

msm.iptw <- cvMSM(y=y, a=a, v=w1, w=w, data=obs.data,
model.msm=list(Size=3, Order=c(1,2), Int=1),
model.av=list(Model=list(c(1))),
model.aw=list(Model=NULL, Size=3,Int=2),
mapping='IPTW', fitting='IPTW', stable.wt=T)
```

Result:

$g(A|W)$:

```
CV selects: size =  2 , interactions =  2
with min.risk: 0.5584379

$Formula [1] "Intercept +  w1 + w2"

$Coefficients
(Intercept)          w1          w2
   1.204914   -1.356494    1.080563
```

MSM $E(Y_a|a, V)$

```
CV selects: size =  2 with min.risk:  1.056349
IPTW estimator:
$Formula [1] "Intercept +  a + w1"

$Coefficients
```

```
(Intercept)           a          w1
  1.5134810   1.9898810   0.9660859
```

R-package cvDSA

- Example 4. Checking the Experimental Treatment Assignment assumption. (No ETA violation)

  Code:

```
obs.data.ETA <- check.ETA(y=y, a=a, v=w1, w=cbind(w1,w2),
data=obs.data, yfamily='gaussian', afamily='binomial',
model.msm=list(Model=list(c(1,0),c(0,1))),
model.aw=list(Model=list(c(1,0),c(0,1))),
model.av=list(Model=list(c(1))),
model.yaw=list(Model=list(c(1,0,0),c(0,1,0),c(0,0,1),c(0,1,1))),
model.yyaw=list(Size=5, Int=2), accuracy=1e-5, stable.wt=F,
n.b=1000, n.sim=100, index.v.inW=c(1))
```

R-package cvDSA

- Example 5. Checking the Experimental Treatment Assignment assumption. (With ETA violations)

  Code:

```
n<-2000;
w1<-runif(n);w2<-runif(n); w3<-runif(n); w4<-runif(n);
w<-cbind(w1,w2,w3,w4);

# Let g(A|W) = logit^(-1) (-1 + w1 - w2 + w1*w3)
p.vec <- diag(4)
model.aw <- list(formula = list(p.vec[1,], p.vec[2,],
p.vec[1,]+p.vec[3,]), coef = c(-1, 1, -5, 1))
        # about 60% violations

# Let E(Y|A, W)=-1+A+w1+w2+w1*w3;
```

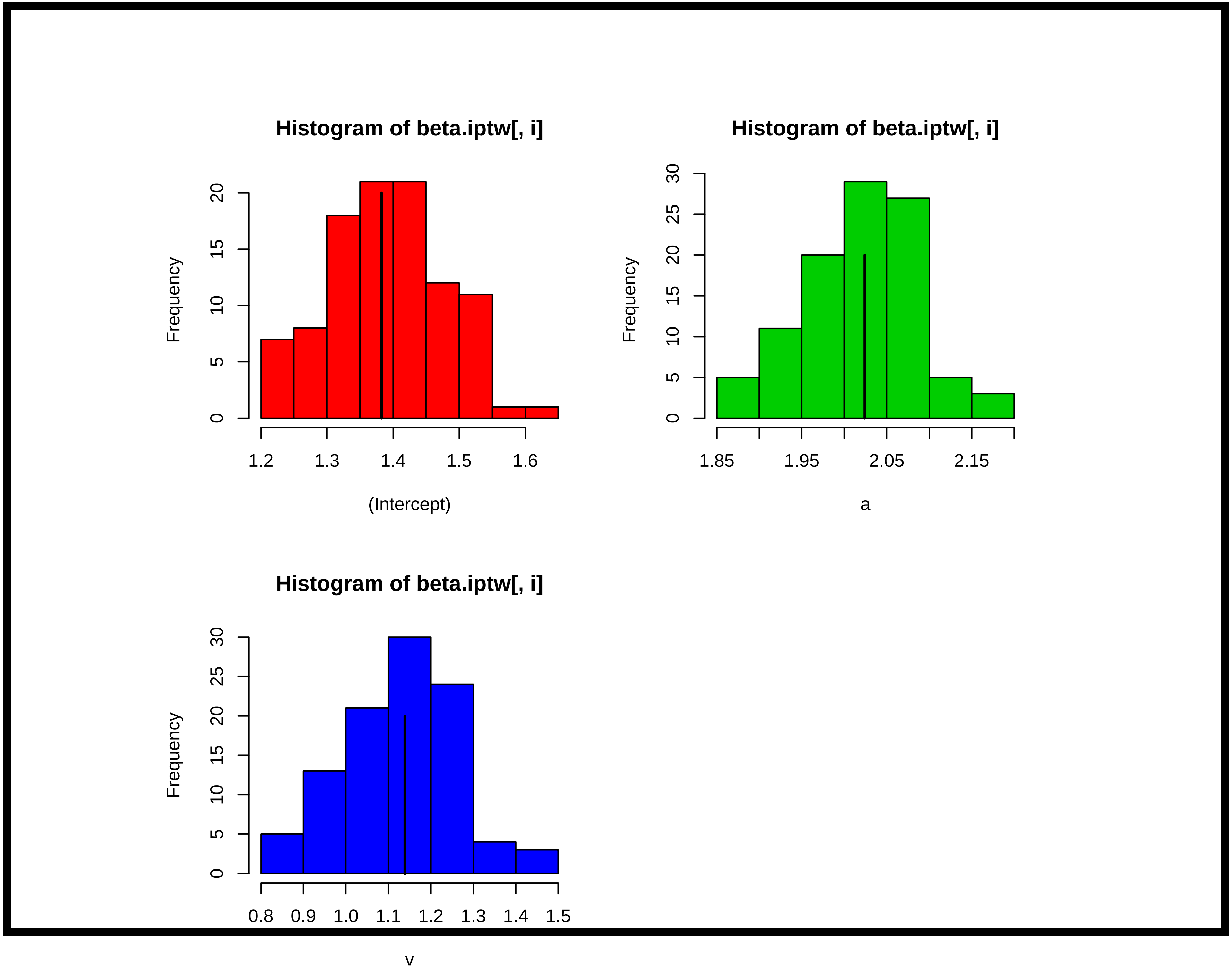
```
p.vec <- diag(5)
model.yaw <-
list(formula=list(p.vec[1,],p.vec[2,],p.vec[3,],
p.vec[2,]+p.vec[4,]), coef=c(-1, 1, 1, 1, 1));

obs.data <- create.obs.data(w, afamily='binomial',
yfamily='gaussian', model.yaw, model.aw)

obs.data.ETA <- check.ETA(y=y, a=a, v=w1, w=w, data=obs.data,
yfamily='gaussian', afamily='binomial',
model.msm=list(Model=list(c(1,0),c(0,1))),
model.aw=list(Model=model.aw$formula),
model.av=list(Model=list(c(1))),
model.yaw=list(Model=model.yaw$formula), wt.censor=NULL,
ncv=5, ncv.nuisance=5, stable.wt=F, fixed.terms=NULL,
cv.risk=F, n.b=1000, n.sim=100, index.v.inW=c(1))
```

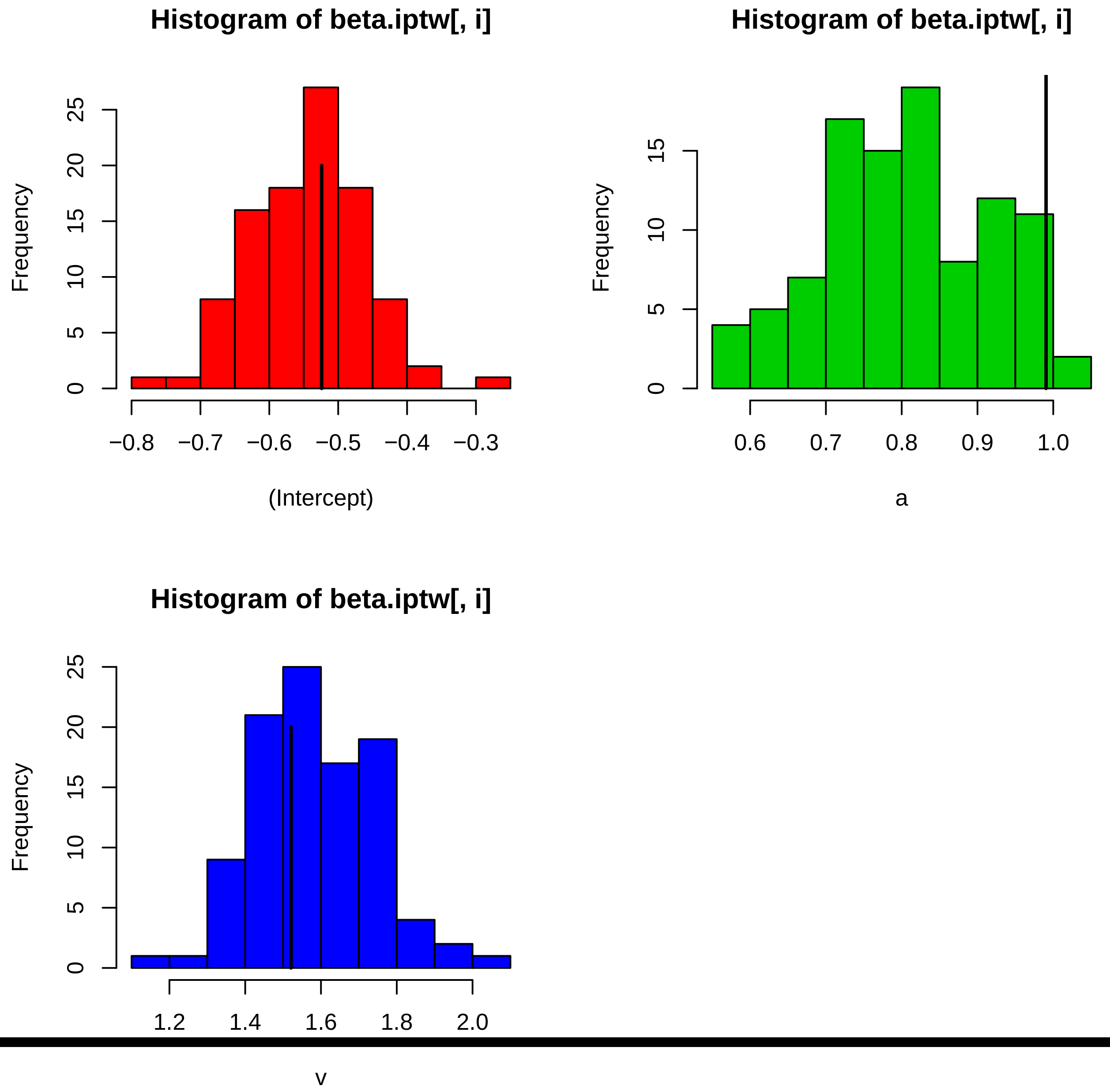
`check.ETA()`

Bootstrap distribution of IPTW causal coefficients: Without ETA violations


Histogram of beta.iptw[, i]
Frequency
(Intercept)
Histogram of beta.iptw[, i]
Frequency
a
Histogram of beta.iptw[, i]
Frequency
v

`check.ETA()`

Bootstrap distribution of IPTW causal coefficients: With ETA violations


Histogram of beta.iptw[, i]
Frequency
0
5
10
15
20
25
−0.8
−0.7
−0.6
−0.5
−0.4
−0.3
(Intercept)
Histogram of beta.iptw[, i]
Frequency
0
5
10
15
0.6
0.7
0.8
0.9
1.0
a
Histogram of beta.iptw[, i]
Frequency
0
5
10
15
20
25
1.2
1.4
1.6
1.8
2.0
v