

Supplementary Web Appendix for: Complier stochastic direct effects: identification and robust estimation

Kara E. Rudolph, Oleg Sofrygin, Mark J. van der Laan

1 Identification proof

Proof.

$$\begin{aligned}
\Psi_{CSDE}(P) &\equiv \{E_0(E_0(E_{g_{M|0,W}^*}\{E_0(Y|W, Z, M)|W, Z\}W, A = 1)|W) \\
&\quad - E_0(E_0(E_{g_{M|0,W}^*}\{E_0(Y|W, Z, M)|W, Z\}W, A = 0)|W)\} \\
&\quad / \{E_0(E_0(Z|W, A = 1)|W) - E_0(E_0(Z|W, A = 0)|W)\} \\
\text{By assumption 1, } &P(Z = z | W, A = a) = P(Z_a = z | W), \text{ so} \\
&\equiv \{E_0(E_0(E_{g_{M|0,W}^*}\{E_0(Y_{g_{M|0,W}^*} | W, Z)\} | W, Z_1) | W) \\
&\quad - E_0(E_0(E_{g_{M|0,W}^*}\{E_0(Y_{g_{M|0,W}^*} | W, Z)\} | W, Z_0) | W)\} \\
&\quad / [E_0\{E_0(Z_1|W) - E_0(Z_0|W)\}] \\
&\equiv \{E_0(E_0(Y_{1,g_{M|0,W}^*}|W) - E_0(Y_{0,g_{M|0,W}^*}|W))\} \\
&\quad / \{E_0(E_0(Z_1|W) - E_0(Z_0|W))\} \\
&\equiv \frac{E_0(Y_{1,g_{M|0,W}^*} - Y_{0,g_{M|0,W}^*})}{E_0(Z_1 - Z_0)} \\
&\equiv \{E_0(Y_{1,g_{M|0,W}^*} - Y_{0,g_{M|0,W}^*}|Z_1 - Z_0 = 1)P(Z_1 - Z_0 = 1) \\
&\quad + E_0(Y_{1,g_{M|0,W}^*} - Y_{0,g_{M|0,W}^*}|Z_1 - Z_0 = 0)P(Z_1 - Z_0 = 0) \\
&\quad + E_0(Y_{1,g_{M|0,W}^*} - Y_{0,g_{M|0,W}^*}|Z_1 - Z_0 = -1)P(Z_1 - Z_0 = -1)\} \\
&\quad / E_0(Z_1 - Z_0) \\
\text{By assumption 2, } & \\
&\equiv \{E_0(Y_{1,g_{M|0,W}^*} - Y_{0,g_{M|0,W}^*}|Z_1 - Z_0 = 1)P(Z_1 - Z_0 = 1) \\
&\quad + E_0(Y_{1,g_{M|0,W}^*} - Y_{0,g_{M|0,W}^*}|Z_1 - Z_0 = -1)P(Z_1 - Z_0 = -1)\} \\
&\quad / E_0(Z_1 - Z_0) \\
\text{By assumption 3, } & \\
&\equiv \{E_0(Y_{1,g_{M|0,W}^*} - Y_{0,g_{M|0,W}^*}|Z_1 - Z_0 = 1)P(Z_1 - Z_0 = 1)\} \\
&\quad / E_0(Z_1 - Z_0) \\
&\quad Z_1 - Z_0 \in \{0, 1\}, \text{ so} \\
&\equiv \{E_0(Y_{1,g_{M|0,W}^*} - Y_{0,g_{M|0,W}^*}|Z_1 - Z_0 = 1)E(Z_1 - Z_0)\} \\
&\quad / E_0(Z_1 - Z_0) \\
&\equiv E_0(Y_{1,g_{M|0,W}^*} - Y_{0,g_{M|0,W}^*}|Z_1 - Z_0 = 1)
\end{aligned}$$

By assumptions 4 and 5, we have that Ψ_{CSDE} is defined. \square

2 Estimator modifications when there is also a direct effect of A on M

The complier stochastic direct effect estimand and estimation approaches we consider also work in the scenario where M may depend on A conditional on Z : $M = f(W, A, Z, U_M)$.

We describe differences in the estimator details for such a scenario here. In this alternative scenario, A is not an instrument for the total effect of Z on Y , and the estimation approach suggested by Joffe et al. (2008) would also be appropriate.

The true distribution P_0 of O can be factorized as

$$P_0(O) = P_0(Y|W, Z, M)P_0(M|W, A, Z)P_0(Z|W, A)P_0(A|W)P_0(W).$$

2.1 Inverse Probability of Treatment Weighted Estimator

The inverse probability of treatment weights for estimating Ψ_{SDE} are

$$IPTW_{SDE} = \frac{(2A - 1)\hat{g}_{M|0,W}}{g_{A|W}g_{M|Z,A,W}}. \quad (1)$$

Let $g_{A,n}$ and $g_{M,n}$ be estimators of $g_{A|W} = P(A = a|W)$ and $g_{M|Z,A,W} = P(M = m|Z, A, W)$, respectively. $g_{A,n}$ can be estimated by predicted probabilities from a logistic regression model of A on W . One could use machine learning in model fitting but we will describe estimation in terms of parametric model fitting for simplicity. $g_{M,n}$ can be estimated by predicted probabilities from a logistic regression model of M on W, A, Z . $\hat{g}_{M|0,W}$ is treated as known, estimated from the observed data, marginalizing out Z : $\sum_{z=0}^1 P(M = m|Z = z, A = 0, W)P(Z = z|A = 0, W)$ (VanderWeele and Tchetgen Tchetgen, 2017). The IPTW estimate of Ψ_{SDE} is the empirical mean of outcome, Y , weighted by an estimate of $IPTW_{SDE}$.

The inverse probability of treatment weights for estimating Ψ_{FS} are as written in the main text.

The associated variance can be estimated as the sample variance of the estimator's influence curve (IC), which is

$$D_{IPTW}(P) = \frac{D_{IPTW_{SDE}}(P)}{\Psi_{FS}(P)} - \frac{\Psi_{SDE}(P)D_{IPTW_{FS}}(P)}{\Psi_{FS}^2(P)}, \quad (2)$$

and where

$$D_{IPTW_{SDE}}(P) = \frac{(2A - 1)\hat{g}_{M|0,W}}{g_{A|W}g_{M|Z,A,W}}Y - \Psi_{SDE} \quad (3)$$

and where

$$D_{IPTW_{FS}}(P) = \frac{2A - 1}{g_{A|W}}Z - \Psi_{FS}. \quad (4)$$

2.2 Estimating Equation Estimator

This estimator solves the efficient influence curve (EIC) for Ψ_{CSDE} , which is given by

$$D_{CSDE}(P)(Q_W, g_A, g_Z, \bar{Q}) = \frac{D_{SDE}(P)}{\Psi_{FS}(P)} - \frac{\Psi_{SDE}(P)D_{FS}(P)}{\Psi_{FS}^2(P)}, \quad (5)$$

where

$$\begin{aligned}
D_{SDE}(P) = & \left(\frac{g_{1|W,Z,M}}{g_{1|W}} - \frac{g_{0|W,Z,M}}{g_{0|W}} \right) \frac{\hat{g}_{M|A=0,W}}{g_{M|Z,A,W}} (Y - \bar{Q}_Y(M, Z, W)) \\
& + \frac{2A-1}{g_{A|W}} (\bar{Q}_M(Z=1, W) - \bar{Q}_M(Z=0, W)) (Z - g_Z(1|A, W)) \\
& + (\bar{Q}_Z(A=1, W) - \bar{Q}_Z(A=0, W)) - \Psi_{SDE}
\end{aligned} \tag{6}$$

and where

$$D_{FS}(P) = \frac{2A-1}{g_{A|W}} (Z - g_Z(1|A, W)) + \{(g_Z(A=1, W) - g_Z(A=0, W)) - \Psi_{FS}\}. \tag{7}$$

We first solve D_{SDE} to obtain the EE estimate of Ψ_{SDE} . We calculate the first component of D_{SDE} as follows. Let $g_M = P(M = m|Z, A, W)$, $g_A = P(A = a|W)$, and $g_{A2} = P(A = a|W, Z, M)$. Recall that $\hat{g}_{M|0,W}$ is treated as known, estimated from the observed data, marginalizing out Z : $\sum_{z=0}^1 P(M = m|Z = z, A = 0, W)P(Z = z|A = 0, W)$ (VanderWeele and Tchetgen Tchetgen, 2017). $g_{M,n}$ can be estimated by predicted probabilities from a logistic regression model of M on Z , A , and W . g_{A2} can be written $\frac{P(A=a|W)P(Z|a,W)P(M|Z,a,W)}{P(M,Z|W)} = \frac{g_{A|W}g_{Z|A,W}g_{M|Z,A,W}}{P(M,Z|W)}$, where $g_{A,n}$ and $g_{M,n}$ can be estimated as described above, $g_{Z,n}$ can be estimated from a logistic regression model of Z on A and W , and where an estimate of $P(Z, M|W)$ is obtained by marginalizing out A : $\left(\sum_{a=0}^1 P(M = m|Z, A = a, W)P(A = a|W) \right) \left(\sum_{a=0}^1 P(Z = z|A = a, W)P(A = a|W) \right)$, which can be rewritten in terms of the above estimators $\sum_{a=0}^1 g_{M,n}g_{A,n} \sum_{a=0}^1 g_{Z,n}g_{A,n}$. The other components can be calculated as described in the main text.

The second and third components of D_{SDE} and the components of D_{FS} are calculated as described in the main text. The associated variance can be estimated as the sample variance of the EIC, $D_{CSDE}(P)$, which is given in Equation 5.

2.3 Compatible Targeted Minimum Loss-Based Estimator

Recall $\bar{Q}_Y = E(Y|W, Z, M)$, $g_M = P(M = m|Z, A, W)$, $g_A = P(A = a|W)$, and $g_{A2} = P(A = a|W, Z, M)$. Again, $\hat{g}_{M|0,W}$ is treated as known, estimated from the observed data, marginalizing out Z : $\sum_{z=0}^1 P(M = m|Z = z, A = 0, W)P(Z = z|A = 0, W)$ (VanderWeele and Tchetgen Tchetgen, 2017). Consider submodel $\{\bar{Q}_{Y,n}(M, Z, W)(\epsilon) : \epsilon\}$ defined as: $\text{logit}(\bar{Q}_{Y,n}(\epsilon)(M, Z, W)) = \text{logit}(\bar{Q}_{Y,n}(M, Z, W)) + \epsilon C_Y$, where $C_Y = \left(\frac{g_{1|W,Z,M}}{g_{1|W}} - \frac{g_{0|W,Z,M}}{g_{0|W}} \right) \frac{\hat{g}_{M|A=0,W}}{g_{M|Z,A,W}}$.

The components of C_Y can be calculated as described in the above subsections and in the main text. The update step for \bar{Q}_Y and the remaining steps for the TMLE estimator are completed as in the main text.

The TMLE solves the efficient influence curve (EIC) for Ψ_{CSDE} (shown in the previous subsection), replacing g_Z and \bar{Q}_Y with g_Z^* and \bar{Q}_Y^* . The variance of the TMLE estimator of Ψ_{CSDE} is estimated as the sample variance of $D_{CSDE}(P)$.

3 R code

3.1 Code for ratio of Inverse Probability of Treatment Weighted Estimators

```

1 #This estimates the complier stochastic direct effect and its variance. It
  takes the following arguments:
2 # a is the instrument, 0/1. It is assumed to be exogenous, but the code can
  be modified to make it conditionally random.
3 # z is the exposure influenced by the instrument, 0/1. It is a function of
  a and w
4 # m is the mediator, 0/1. It is a function of z, w.
5 # y is the outcome, 0/1, but the code can be modified for any outcome type.
  It is a function of z, w, m.
6 # w is a matrix of covariates
7 # svywt is a vector of weights to be applied to the data.
8 # zmodel is the parametric model for z.
9 # mmodel is the parametric model for m.
10 # ymodel is the parametric model for y.
11 # qmodel is the parametric model for q.
12 # gm is the user-specified stochastic intervention on M, conditional on a=0
  and w
13 # za, za1, and za0 are optional arguments that can be included if the user
  estimates these as part of the stochastic intervention. Otherwise, they
  are estimated within the function
14 # uses the constrained regression function if za, za1, and za0 are null
15
16 mediptw<-function(a, z, m, y, w, svywt, zmodel, mmodel, ymodel, qmodel, gm,
  za=NULL, za1=NULL, za0=NULL){
17
18 datw<-w
19
20 # estimate p(m | w, z)
21 mz<-predict(glm(formula=mmodel, family="binomial", data=data.frame(cbind(
  datw, z=z, m=m))), newdata=data.frame(cbind(datw, z=z)), type="response
  ")
22 mz0<-predict(glm(formula=mmodel, family="binomial", data=data.frame(cbind(
  datw, z=z, m=m))), newdata=data.frame(cbind(datw, z=0)), type="response
  ")
23 mz1<-predict(glm(formula=mmodel, family="binomial", data=data.frame(cbind(
  datw, z=z, m=m))), newdata=data.frame(cbind(datw, z=1)), type="response
  ")
24
25 # estimate p(z | w, a)
26 if(is.null(za) | is.null(za1) | is.null(za0)){
27 zfit<-mle.logreg.constrained(formula(zmodel), data.frame(cbind(datw, a=a,
  z=z)))
28

```

```

29   za0<-predictClogis(cbind(rep(0,nrow(data.frame(datw))), datw), zfit$beta)
30   zal<-predictClogis(cbind(rep(1,nrow(data.frame(datw))), datw), zfit$beta)
31   za<-predictClogis(data.frame(cbind(a=a, datw)), zfit$beta)
32 }
33 else {
34   za<-za
35   zal<-zal
36   za0<-za0
37 }
38
39 pzal<-ifelse(z==1, zal, 1-zal)
40 pza0<-ifelse(z==1, za0, 1-za0)
41
42 # estimate p(a/w,m,z) using previous estimates. Note that p(a/w,m,z) = p(a/
    w,z) bc of exclusion restriction
43 pal<-(mean(a)*pzal)/(pzal*mean(a) + pza0*mean(1-a))
44 palz0<-(mean(a)*(1-zal))/((1-zal)*mean(a) + (1-za0)*mean(1-a))
45 palz1<-(mean(a)*zal)/(zal*mean(a) + za0*mean(1-a))
46
47 tmpdat<-data.frame(cbind(datw, a=a))
48
49 #make clever covariate
50 psm<- (mz*m) + ((1-mz)*(1-m))
51
52 tmpdat$wts<-((m*gm + (1-m)*(1-gm))/psm)* svywt
53 #component that can't go into the weights
54 tmpdat$cc<- (pal/mean(a)) - ((1-pal)/mean(1-a))
55
56 tmpdat$ccz0<- (palz0/mean(a)) - ((1-palz0)/mean(1-a))
57 tmpdat$ccz1<- (palz1/mean(a)) - ((1-palz1)/mean(1-a))
58
59 tmpdat$y<-y
60
61 psi1<-sum(tmpdat$y * tmpdat$wts * tmpdat$cc)/sum(svywt)
62 eicpsi1<-(tmpdat$cc*tmpdat$wts * tmpdat$y) - psi1
63
64 #estimate denominator
65 psi2<-sum(z * tmpdat$cc *svywt)/sum(svywt)
66 eicpsi2<-(z * tmpdat$cc * svywt) - psi2
67
68 csde<-psi1/psi2
69 csdeec<-(eicpsi1/psi2) - ((psi1*eicpsi2)/(psi2^2))
70 varcsde<-var(csdeec)/nrow(tmpdat)
71
72 return(list("est"=csde, "var"=varcsde))
73 }

```

CSDE_iptw.R

3.2 Code for ratio of Estimating Equation Estimators

```

1 #This estimates the complier stochastic direct effect and its variance. It
2 takes the following arguments:
3 # a is the instrument, 0/1. It is assumed to be exogenous, but the code can
4 be modified to make it conditionally random.
5 # z is the exposure influenced by the instrument, 0/1. It is a function of
6 a and w
7 # m is the mediator, 0/1. It is a function of z, w.
8 # y is the outcome, 0/1, but the code can be modified for any outcome type.
9 It is a function of z, w, m.
10 # w is a matrix of covariates
11 # svywt is a vector of weights to be applied to the data.
12 # zmodel is the parametric model for z.
13 # mmodel is the parametric model for m.
14 # ymodel is the parametric model for y.
15 # qmodel is the parametric model for q.
16 # gm is the user-specified stochastic intervention on M, conditional on a=0
17 and w
18 # za, za1, and za0 are optional arguments that can be included if the user
19 estimates these as part of the stochastic intervention. Otherwise, they
20 are estimated within the function
21 # uses the constrained regression function if za, za1, and za0 are null
22
23 medee<-function(a, z, m, y, w, svywt, zmodel, mmodel, ymodel, qmodel, gm,
24   za=NULL, za1=NULL, za0=NULL){
25
26   datw<-w
27
28   # estimate p(m | w, z)
29   mz<-predict(glm(formula=mmodel, family="binomial", data=data.frame(cbind(
30     datw, z=z, m=m))), newdata=data.frame(cbind(datw, z=z)), type="response")
31
32   mz0<-predict(glm(formula=mmodel, family="binomial", data=data.frame(cbind(
33     datw, z=z, m=m))), newdata=data.frame(cbind(datw, z=0)), type="response")
34
35   mz1<-predict(glm(formula=mmodel, family="binomial", data=data.frame(cbind(
36     datw, z=z, m=m))), newdata=data.frame(cbind(datw, z=1)), type="response")
37
38   # estimate p(z | w, a)
39   if(is.null(za) | is.null(za1) | is.null(za0)){
40     zfit<-mle.logreg.constrained(formula(zmodel), data.frame(cbind(datw, a=a,
41       z=z)))
42
43     za0<-predictClogis(cbind(rep(0,nrow(data.frame(datw))), datw), zfit$beta)
44     za1<-predictClogis(cbind(rep(1,nrow(data.frame(datw))), datw), zfit$beta)
45     za<-predictClogis(data.frame(cbind(a=a, datw)), zfit$beta)
46   }
47   else {
48     za<-za
49     za1<-za1
50     za0<-za0
51   }
52 }

```

```

39 pzal<-ifelse(z==1, zal, 1-zal)
40 pza0<-ifelse(z==1, za0, 1-za0)
41
42 # estimate  $p(a/w, m, z)$  using previous estimates. Note that  $p(a/w, m, z) = p(a/$ 
     $w, z)$  bc of exclusion restriction
43 pal<-(mean(a)*pzal)/(pzal*mean(a) + pza0*mean(1-a))
44 palz0<-(mean(a)*(1-zal))/((1-zal)*mean(a) + (1-za0)*mean(1-a))
45 palz1<-(mean(a)*zal)/(zal*mean(a) + za0*mean(1-a))
46
47 tmpdat<-data.frame(cbind(datw, a=a))
48
49 #get initial Y fit
50 yfit<-glm(formula=ymodel, family="binomial", data=data.frame(cbind(datw,
    z=z, m=m, y=y)))
51 tmpdat$yinit<-cbind(predict(yfit, newdata=data.frame(cbind(datw, z=z, m=
    m))), type="response"),
52   predict(yfit, newdata=data.frame(cbind(datw, z=z, m=1))), type="response
    "),
53   predict(yfit, newdata=data.frame(cbind(datw, z=z, m=0))), type="response
    ") )
54 tmpdat$yinitz0<-cbind(predict(yfit, newdata=data.frame(cbind(datw, z=0,
    m=1))), type="response"), predict(yfit, newdata=data.frame(cbind(datw,
    z=0, m=0))), type="response" )
55 tmpdat$yinitz1<-cbind(predict(yfit, newdata=data.frame(cbind(datw, z=1,
    m=1))), type="response"), predict(yfit, newdata=data.frame(cbind(datw,
    z=1, m=0))), type="response" )
56
57 #make clever covariate
58 psm<-(m*z) + ((1-mz)*(1-m))
59
60 tmpdat$wts<-((m*gm + (1-m)*(1-gm))/psm)* svywt
61 #component that can't go into the weights
62 tmpdat$cc<- (pal/mean(a)) - ((1-pal)/mean(1-a))
63
64 tmpdat$ccz0<- (palz0/mean(a)) - ((1-palz0)/mean(1-a))
65 tmpdat$ccz1<- (palz1/mean(a)) - ((1-palz1)/mean(1-a))
66
67 tmpdat$y<-y
68
69 epsilon<-coef(glm(y ~ -1 + offset(qlogis(yinit[,1])) +cc, weights=wts,
    family="quasibinomial", data=tmpdat)) #
70
71 eic1<-(tmpdat$cc)*tmpdat$wts * (tmpdat$y - tmpdat$yinit[,1])
72
73 #integrate out M to get qm
74 tmpdat$qm<-tmpdat$yinit[,2]*gm + tmpdat$yinit[,3]*(1-gm)
75 tmpdat$qmz0<-tmpdat$yinitz0[,1]*gm + tmpdat$yinitz0[,2]*(1-gm)
76 tmpdat$qmz1<-tmpdat$yinitz1[,1]*gm + tmpdat$yinitz1[,2]*(1-gm)
77
78 #initial fit gz
79 gz<-cbind(za, za0, zal)
80
81 #make components for second targeting step

```



```

82 tmpdat$difqma<-tmpdat$qmz1 - tmpdat$qmz0
83
84 tmpdat$ga<-ifelse(a==1, mean(a), mean(1-a))
85 tmpdat$a<-a
86 tmpdat$nota<-1-a
87
88 #integrate out z to get qz
89 qz<-cbind((tmpdat$qmz1*gz[,2]) + (tmpdat$qmz0*(1-gz[,2])), (tmpdat$qmz1*
90   gz[,3]) + (tmpdat$qmz0*(1-gz[,3])))
91
92 #estimate numerator
93 eic2<-((2*a-1)/tmpdat$ga)*svywt*(tmpdat$qmz1 - tmpdat$qmz0) *(z-gz[,1])
94 #eic2<-(tmpdat$a*svywt*(tmpdat$qmz1 - tmpdat$qmz0) + tmpdat$nota*svywt*(
95   tmpdat$qmz1 - tmpdat$qmz0))* (z-gz[,1])
96
97 #estimate denominator
98 eic3<-((qz[,2] - qz[,1])*svywt)
99 psil<-mean(eic1 + eic2+ eic3)
100 eicdp1<-eic1 + eic2 + eic3 -psil
101
102 eic1dp2<-((2*a-1)/tmpdat$ga)*svywt*(z - gz[,1])
103 eic2dp2<-((gz[,3] - gz[,2])*svywt)
104 psi2<-mean(eic1dp2 + eic2dp2)
105 eicdp2<-eic1dp2 + eic2dp2 - psi2
106
107 csde<-psil/psi2
108 csdeec<-(eicdp1/psi2) - ((psil*eicdp2)/(psi2^2))
109 varcsde<-var(csdeec)/nrow(tmpdat)
110
111 return(list("est"=csde, "var"=varcsde))
112 }

```

CSDE_ee.R

3.3 Code for ratio of Targeted Minimum Loss-based Estimators (Efficient TMLE)

```

1 #This estimates the complier stochastic direct effect and its variance. It
2 takes the following arguments:
3 # a is the instrument, 0/1. It is assumed to be exogenous, but the code can
4 be modified to make it conditionally random.
5 # z is the exposure influenced by the instrument, 0/1. It is a function of
6 a and w
7 # m is the mediator, 0/1. It is a function of z, w.
8 # y is the outcome, 0/1, but the code can be modified for any outcome type.
9 It is a function of z, w, m.
10 # w is a matrix of covariates
11 # svywt is a vector of weights to be applied to the data.
12 # zmodel is the parametric model for z.
13 # mmodel is the parametric model for m.
14 # ymodel is the parametric model for y.
15 # qmodel is the parametric model for q.

```

```

12 # gm is the user-specified stochastic intervention on M, conditional on a=0
    and w
13 # za, za1, and za0 are optional arguments that can be included if the user
    estimates these as part of the stochastic intervention. Otherwise, they
    are estimated within the function
14 # uses the constrained regression function if za, za1, and za0 are null
15
16 medtmle<-function(a, z, m, y, w, svywt, zmodel, mmodel, ymodel, qmodel, gm,
    za=NULL, za1=NULL, za0=NULL){
17
18   datw<-w
19
20   # estimate p(m | w, z)
21   mz<-predict(glm(formula=mmodel, family="binomial", data=data.frame(cbind(
    datw, z=z, m=m))), newdata=data.frame(cbind(datw, z=z)), type="response
    ")
22   mz0<-predict(glm(formula=mmodel, family="binomial", data=data.frame(cbind(
    datw, z=z, m=m))), newdata=data.frame(cbind(datw, z=0)), type="response
    ")
23   mz1<-predict(glm(formula=mmodel, family="binomial", data=data.frame(cbind(
    datw, z=z, m=m))), newdata=data.frame(cbind(datw, z=1)), type="response
    ")
24
25   # estimate p(z | w, a)
26   if(is.null(za) | is.null(za1) | is.null(za0)){
27     zfit<-mle.logreg.constrained(formula(zmodel), data.frame(cbind(datw, a=a,
    z=z)))
28
29     za0<-predictClogis(cbind(rep(0,nrow(data.frame(datw))), datw), zfit$beta)
30     za1<-predictClogis(cbind(rep(1,nrow(data.frame(datw))), datw), zfit$beta)
31     za<-predictClogis(data.frame(cbind(a=a, datw)), zfit$beta)
32   }
33   else {
34     za<-za
35     za1<-za1
36     za0<-za0
37   }
38
39   pza1<-ifelse(z==1, za1, 1-za1)
40   pza0<-ifelse(z==1, za0, 1-za0)
41
42   # estimate p(a/w,m,z) using previous estimates. Note that p(a/w,m,z) = p(a/
    w,z) bc of exclusion restriction
43   pal<-(mean(a)*pza1)/(pza1*mean(a) + pza0*mean(1-a))
44   palz0<-(mean(a)*(1-za1))/((1-za1)*mean(a) + (1-za0)*mean(1-a))
45   palz1<-(mean(a)*za1)/(za1*mean(a) + za0*mean(1-a))
46
47   tmpdat<-data.frame(cbind(datw, a=a))
48
49   #get initial Y fit
50   yfit<-glm(formula=ymodel, family="binomial", data=data.frame(cbind(datw,
    z=z, m=m, y=y)))

```

```

51 tmpdat$yinit<-cbind(predict(yfit , newdata=data.frame(cbind(datw, z=z, m=
52 m)), type="response"),
53 predict(yfit , newdata=data.frame(cbind(datw, z=z, m=1)), type="response
54 ")),
55 predict(yfit , newdata=data.frame(cbind(datw, z=z, m=0)), type="response
56 "))
57 tmpdat$yinitz0<-cbind(predict(yfit , newdata=data.frame(cbind(datw, z=0,
58 m=1)), type="response"), predict(yfit , newdata=data.frame(cbind(datw,
59 z=0, m=0)), type="response"))
60 tmpdat$yinitz1<-cbind(predict(yfit , newdata=data.frame(cbind(datw, z=1,
61 m=1)), type="response"), predict(yfit , newdata=data.frame(cbind(datw,
62 z=1, m=0)), type="response"))
63
64 #make clever covariate
65 psm<-(mz*m) + ((1-mz)*(1-m))
66
67 tmpdat$wts<-((m*gm + (1-m)*(1-gm))/psm)* svywt
68 #component that can't go into the weights
69 tmpdat$cc<- (pal/mean(a)) - ((1-pal)/mean(1-a))
70
71 tmpdat$ccz0<- (palz0/mean(a)) - ((1-palz0)/mean(1-a))
72 tmpdat$ccz1<- (palz1/mean(a)) - ((1-palz1)/mean(1-a))
73
74 tmpdat$y<-y
75
76 epsilon<-coef(glm(y ~ -1 + offset(qlogis(qyinit[,1])) + cc ,weights=wts,
77 family="quasibinomial", data=tmpdat)) #
78 epsilon<-ifelse(is.na(epsilon), 0, epsilon)
79
80 #update Qy
81 tmpdat$yup<-plogis(qlogis(tmpdat$yinit) + epsilon*(tmpdat$cc))
82 tmpdat$yupz0m0<-plogis(qlogis(tmpdat$yinitz0[,2]) + epsilon * tmpdat$
83 ccz0)
84 tmpdat$yupz0m1<-plogis(qlogis(tmpdat$yinitz0[,1]) + epsilon * tmpdat$
85 ccz0)
86 tmpdat$yupz1m0<-plogis(qlogis(tmpdat$yinitz1[,2]) + epsilon * tmpdat$
87 ccz1)
88 tmpdat$yupz1m1<-plogis(qlogis(tmpdat$yinitz1[,1]) + epsilon * tmpdat$
89 ccz1)
90
91 eic1<-(tmpdat$cc)*tmpdat$wts * (tmpdat$y - tmpdat$yup[,1])
92
93 #integrate out M to get qm
94 tmpdat$qm<-tmpdat$yup[,2]*gm + tmpdat$yup[,3]*(1-gm)
95 tmpdat$qmz0<-tmpdat$yupz0m1*gm + tmpdat$yupz0m0*(1-gm)
96 tmpdat$qmz1<-tmpdat$yupz1m1*gm + tmpdat$yupz1m0*(1-gm)
97
98 #initial fit gz
99 gz<-cbind(za, za0, za1)
100
101 #make components for second targeting step
102 tmpdat$difqmza<-tmpdat$qmz1 - tmpdat$qmz0
103

```

```

92 tmpdat$ga<-ifelse(a==1, mean(a), mean(1-a))
93 tmpdat$a<-a
94 tmpdat$nota<-1-a
95
96 fitcz<- glm(z ~ -1 + a:difqma + nota:difqma, weights=svywt*(1/tmpdat$ga
97 ), family="quasibinomial", data=tmpdat, offset=qlogis(gz[,1]))
98
99 epsiloncz<-coef(fitcz)
100
101 #update gz
102 gzup<-cbind(plogis(qlogis(gz[,1]) + I(tmpdat$a==0)*epsiloncz[2]*tmpdat$
103   difqma + I(tmpdat$a==1)*epsiloncz[1]*tmpdat$difqma),
104   plogis(qlogis(gz[,2]) + epsiloncz[2]*tmpdat$difqma),
105   plogis(qlogis(gz[,3]) + epsiloncz[1]*tmpdat$difqma)
106 )
107
108 #integrate out z to get qz
109 qz<-cbind((tmpdat$qmz1*gzup[,2]) + (tmpdat$qmz0*(1-gzup[,2])), (tmpdat$
110   qmz1*gzup[,3]) + (tmpdat$qmz0*(1-gzup[,3])))
111
112 #estimate numerator
113 psi1<-sum((qz[,2] - qz[,1])*svywt)/sum(svywt)
114
115 #eic2<-(tmpdat$a*svywt*(tmpdat$qmz1 - tmpdat$qmz0) + tmpdat$nota*svywt*(
116   tmpdat$qmz1 - tmpdat$qmz0))*(z-gzup[,1])
117 eic2<-((2*tmpdat$a-1)/tmpdat$ga)*svywt*(tmpdat$qmz1 - tmpdat$qmz0)*(z-
118   gzup[,1])
119 #target gz for denominator
120 fitczd<- glm(z ~ -1 + a + nota, weights=svywt*(1/tmpdat$ga), family="
121   quasibinomial", data=tmpdat, offset=qlogis(gz[,1]))
122 epsilonczd<-coef(fitczd)
123 gzupd<-cbind(plogis(qlogis(gz[,1]) + I(tmpdat$a==0)*epsilonczd[2] + I(
124   tmpdat$a==1)*epsilonczd[1]),
125   plogis(qlogis(gz[,2]) + epsilonczd[2]),
126   plogis(qlogis(gz[,3]) + epsilonczd[1])
127 )
128 #estimate denominator
129 psi2<-sum((gzupd[,3] - gzupd[,2])*svywt)/sum(svywt)
130
131
132 eic3<-((qz[,2] - qz[,1])*svywt) - psi1
133 eicdp1<-eic1 + eic2+ eic3
134
135 eic1dp2<-((2*a-1)/tmpdat$ga)*svywt*(z - gz[,1])
136 eic2dp2<-((gzupd[,3] - gzupd[,2])*svywt) - psi2
137 eicdp2<-eic1dp2 + eic2dp2
138
139 csde<-psi1/psi2
140 csdeec<-((eicdp1/psi2) - ((psi1*eicdp2)/(psi2^2)))
141 varcsde<-var(csdeec)/nrow(tmpdat)
142
143 return(list("est"=csde, "var"=varcsde))
144 }

```

3.4 Code for Targeted Minimum Loss-based Estimator that estimates ratio directly (Compatible TMLE)

```

1 #This estimates the complier stochastic direct effect and its variance. It
2 takes the following arguments:
3 # a is the instrument, 0/1. It is assumed to be exogenous, but the code can
4 be modified to make it conditionally random.
5 # z is the exposure influenced by the instrument, 0/1. It is a function of
6 a and w
7 # m is the mediator, 0/1. It is a function of z, w.
8 # y is the outcome, 0/1, but the code can be modified for any outcome type.
9 It is a function of z, w, m.
10 # w is a matrix of covariates
11 # svywt is a vector of weights to be applied to the data.
12 # zmodel is the parametric model for z.
13 # mmodel is the parametric model for m.
14 # ymodel is the parametric model for y.
15 # qmodel is the parametric model for q.
16 # gm is the user-specified stochastic intervention on M, conditional on a=0
17 and w
18 # za, za1, and za0 are optional arguments that can be included if the user
19 estimates these as part of the stochastic intervention. Otherwise, they
20 are estimated within the function
21 # uses the constrained regression function if za, za1, and za0 are null
22 medtmle<-function(a, z, m, y, w, svywt, zmodel, mmodel, ymodel, qmodel, gm,
23                   za=NULL, za1=NULL, za0=NULL){
24
25   datw<-w
26
27   # estimate p(m | w, z)
28   mz<-predict(glm(formula=mmodel, family="binomial", data=data.frame(cbind(
29     datw, z=z, m=m))), newdata=data.frame(cbind(datw, z=z)), type="response")
30
31   mz0<-predict(glm(formula=mmodel, family="binomial", data=data.frame(cbind(
32     datw, z=z, m=m))), newdata=data.frame(cbind(datw, z=0)), type="response")
33
34   mz1<-predict(glm(formula=mmodel, family="binomial", data=data.frame(cbind(
35     datw, z=z, m=m))), newdata=data.frame(cbind(datw, z=1)), type="response")
36
37   # estimate p(z | w, a)
38   if(is.null(za) | is.null(za1) | is.null(za0)){
39     zfit<-mle.logreg.constrained(formula(zmodel), data.frame(cbind(datw, a=a,
40       z=z)))
41
42     za0<-predictClogis(cbind(rep(0,nrow(data.frame(datw))), datw), zfit$beta)
43     za1<-predictClogis(cbind(rep(1,nrow(data.frame(datw))), datw), zfit$beta)

```

```

31   za<-predictClogis(data.frame(cbind(a=a, datw)), zfit$beta)
32 }
33 else {
34   za<-za
35   zal<-zal
36   za0<-za0
37 }
38
39 pzal<-ifelse(z==1, zal, 1-zal)
40 pza0<-ifelse(z==1, za0, 1-za0)
41
42 # estimate  $p(a/w, m, z)$  using previous estimates. Note that  $p(a/w, m, z) = p(a/$ 
43    $w, z)$  bc of exclusion restriction
43 pal<-(mean(a)*pzal)/(pzal*mean(a) + pza0*mean(1-a))
44 palz0<-(mean(a)*(1-zal))/((1-zal)*mean(a) + (1-za0)*mean(1-a))
45 palz1<-(mean(a)*zal)/(zal*mean(a) + za0*mean(1-a))
46
47 tmpdat<-data.frame(cbind(datw, a=a))
48
49 #get initial Y fit
50 yfit<-glm(formula=ymodel, family="binomial", data=data.frame(cbind(datw,
51   z=z, m=m, y=y)))
51 tmpdat$yinit<-cbind(predict(yfit, newdata=data.frame(cbind(datw, z=z, m=
52   m)), type="response"),
52   predict(yfit, newdata=data.frame(cbind(datw, z=z, m=1)), type="response
53   ")),
53   predict(yfit, newdata=data.frame(cbind(datw, z=z, m=0)), type="response
54   "))
54 tmpdat$yinitz0<-cbind(predict(yfit, newdata=data.frame(cbind(datw, z=0,
55   m=1)), type="response"), predict(yfit, newdata=data.frame(cbind(datw,
56   z=0, m=0)), type="response"))
55 tmpdat$yinitz1<-cbind(predict(yfit, newdata=data.frame(cbind(datw, z=1,
56   m=1)), type="response"), predict(yfit, newdata=data.frame(cbind(datw,
57   z=1, m=0)), type="response"))
56
57 #make clever covariate
58 psm<-(mz*m) + ((1-mz)*(1-m))
59
60 tmpdat$wts<-((m*gm + (1-m)*(1-gm))/psm)* svywt
61 #component that can't go into the weights
62 tmpdat$cc<- (pal/mean(a)) - ((1-pal)/mean(1-a))
63
64 tmpdat$ccz0<- (palz0/mean(a)) - ((1-palz0)/mean(1-a))
65 tmpdat$ccz1<- (palz1/mean(a)) - ((1-palz1)/mean(1-a))
66
67 tmpdat$y<-y
68
69 epsilon<-coef(glm(y ~ -1 + offset(qlogis(yinit[,1])) + cc, weights=wts,
70   family="quasibinomial", data=tmpdat)) #
71
72 #update Qy
72 tmpdat$yup<-plogis(qlogis(tmpdat$yinit) + epsilon*(tmpdat$cc))

```

```

73 tmpdat$qyupz0m0<-plogis(qlogis(tmpdat$qyinitz0[,2]) + epsilon * tmpdat$
    ccz0)
74 tmpdat$qyupz0m1<-plogis(qlogis(tmpdat$qyinitz0[,1]) + epsilon * tmpdat$
    ccz0)
75 tmpdat$qyupz1m0<-plogis(qlogis(tmpdat$qyinitz1[,2]) + epsilon * tmpdat$
    ccz1)
76 tmpdat$qyupz1m1<-plogis(qlogis(tmpdat$qyinitz1[,1]) + epsilon * tmpdat$
    ccz1)
77
78 eic1<-(tmpdat$cc)*tmpdat$wts * (tmpdat$y - tmpdat$qyup[,1])
79
80 #integrate out M to get qm
81 tmpdat$qm<-tmpdat$qyup[,2]*gm + tmpdat$qyup[,3]*(1-gm)
82 tmpdat$qmz0<-tmpdat$qyupz0m1*gm + tmpdat$qyupz0m0*(1-gm)
83 tmpdat$qmz1<-tmpdat$qyupz1m1*gm + tmpdat$qyupz1m0*(1-gm)
84
85 #initial fit gz
86 gz<-cbind(za, za0, za1)
87
88 #make components for second targeting step
89 tmpdat$difqma<-tmpdat$qmz1 - tmpdat$qmz0
90
91 tmpdat$ga<-ifelse(a==1, mean(a), mean(1-a))
92 tmpdat$a<-a
93 tmpdat$nota<-1-a
94
95 fitcz<- glm(z ~ -1 + a + nota + a:difqma + nota:difqma, weights=svywt*
    (1/tmpdat$ga), family="quasibinomial", data=tmpdat, offset=qlogis(gz
    [,1]))
96
97 epsiloncz<-coef(fitcz)
98
99 #update gz
100 gzup<-cbind(plogis(qlogis(gz[,1]) + I(tmpdat$a==0)*epsiloncz[2] + I(
    tmpdat$a==0)*epsiloncz[4]*tmpdat$difqma + I(tmpdat$a==1)*epsiloncz
    [1] + I(tmpdat$a==1)*epsiloncz[3]*tmpdat$difqma),
101   plogis(qlogis(gz[,2]) + epsiloncz[2] + epsiloncz[4]*tmpdat$difqma),
102   plogis(qlogis(gz[,3]) + epsiloncz[1] + epsiloncz[3]*tmpdat$difqma)
103 )
104
105 #integrate out z to get qz
106 qz<-cbind((tmpdat$qmz1*gzup[,2]) + (tmpdat$qmz0*(1-gzup[,2])), (tmpdat$
    qmz1*gzup[,3]) + (tmpdat$qmz0*(1-gzup[,3])))
107
108 #estimate numerator
109 psi1<-sum((qz[,2] - qz[,1])*svywt)/sum(svywt)
110
111 eic2<-((2*a-1)/tmpdat$ga)*svywt*(tmpdat$qmz1 - tmpdat$qmz0) *(z-gzup[,1])
112 #estimate denominator
113 psi2<-sum((gzup[,3] - gzup[,2])*svywt)/sum(svywt)
114
115 eic3<-((qz[,2] - qz[,1])*svywt) - psi1
116 eicdpl<-eic1 + eic2+ eic3

```

```

117
118 eic1dp2<-((2*a-1)/tmpdat$ga)*svywt*(z - gz[,1])
119 eic2dp2<-((gzup[,3] - gzup[,2])*svywt) - psi2
120 eicdp2<-eic1dp2 + eic2dp2
121
122 csde<-psi1/psi2
123 csdeec<-((eicdp1/psi2) - ((psi1*eicdp2)/(psi2^2))
124 varcsde<-var(csdeec)/nrow(tmpdat)
125
126 return(list("est"=csde, "var"=varcsde))
127 }

```

CSDE_tmle.R

4 Figures for stochastic direct and indirect effects

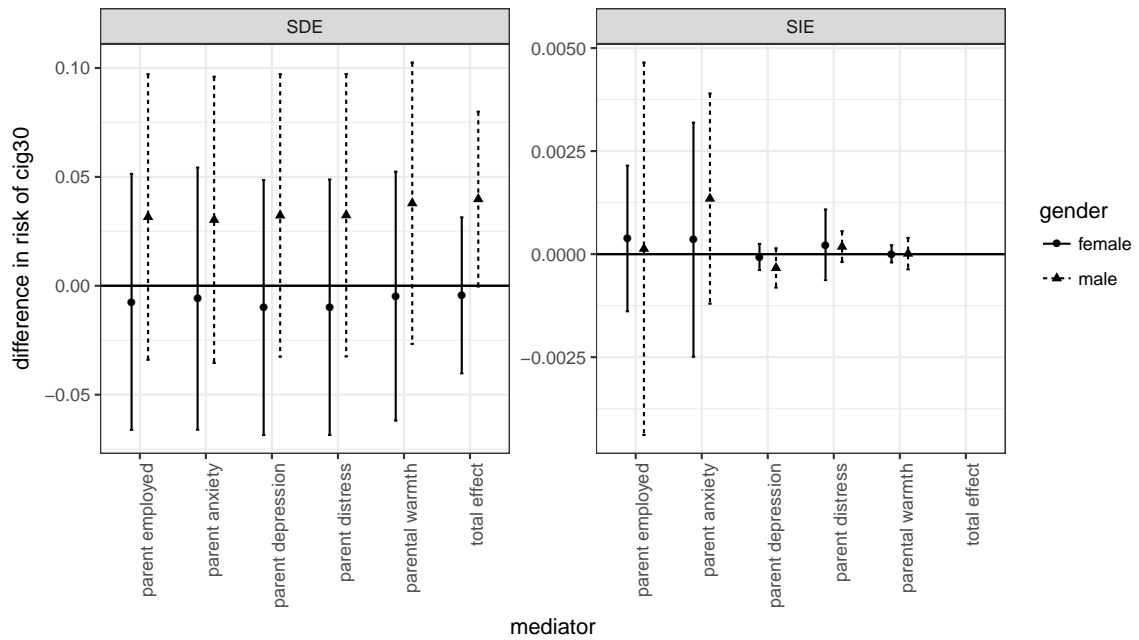


Figure 1: Data-dependent stochastic direct and indirect effect estimates and 95% confidence intervals on past-month cigarette use by mediator. Data from the Moving to Opportunity experiment, interim follow up.

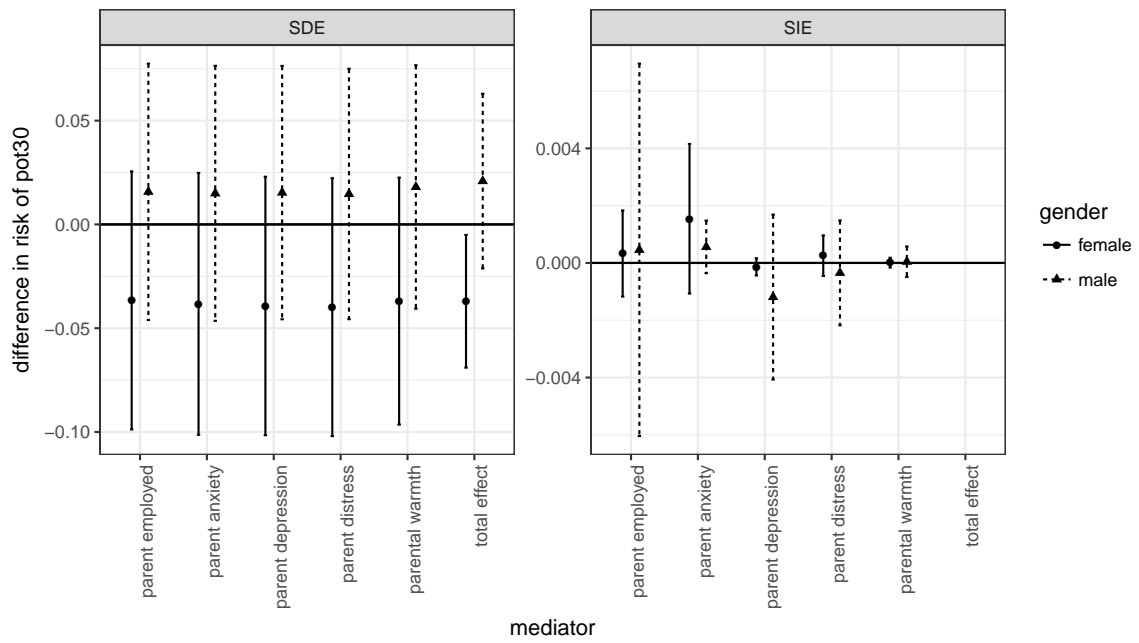


Figure 2: Data-dependent stochastic direct and indirect effect estimates and 95% confidence intervals on past-month marijuana use by mediator. Data from the Moving to Opportunity experiment, interim follow up.

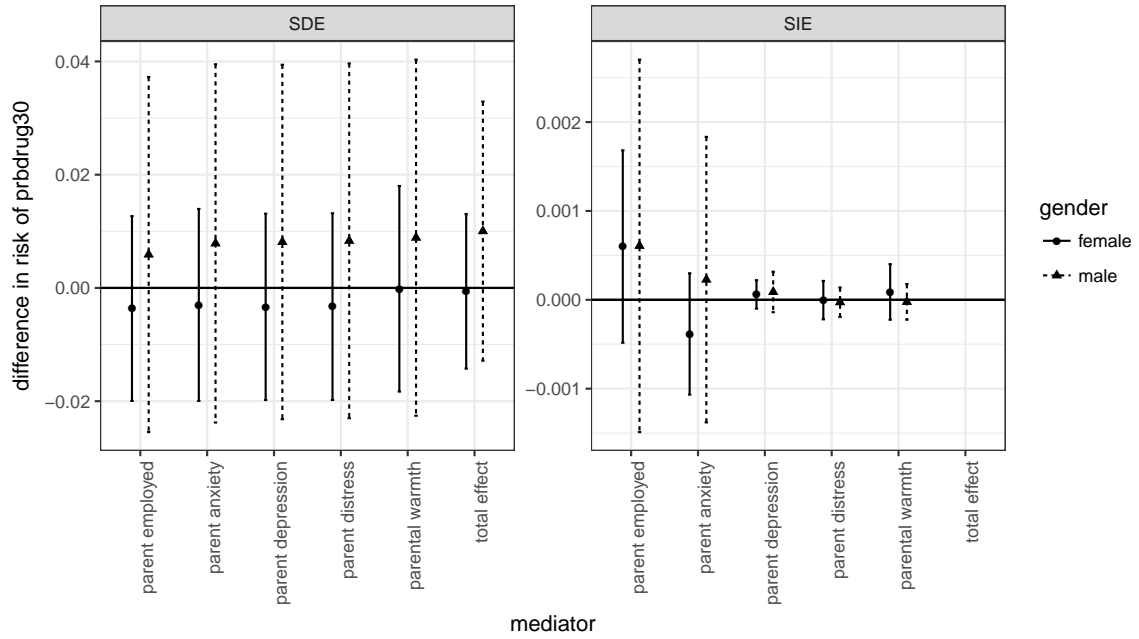


Figure 3: Data-dependent stochastic direct and indirect effect estimates and 95% confidence intervals on past-month problematic drug use by mediator. Data from the Moving to Opportunity experiment, interim follow up.

Bibliography

- Joffe, M. M., Small, D., Ten Have, T., Brunelli, S. and Feldman, H. I. (2008) Extended instrumental variables estimation for overall effects. *The international journal of biostatistics*, **4**.
- VanderWeele, T. J. and Tchetgen Tchetgen, E. J. (2017) Mediation analysis with time varying exposures and mediators. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **79**, 917–938.