[SER 2021 Workshop] Causal Mediation: Modern Methods for Path Analysis

Iván Díaz, Nima Hejazi, Kara Rudolph

updated: May 12, 2021

Contents

Welcome to SER!

This open source, reproducible vignette accompanies a half-day workshop on modern methods for *causal mediation analysis*, given at the SER 2021 Meeting[1] on Monday, 24 May 2021. While we encourage use of this bookdown site, for convenience, we have also made these workshop materials available in PDF[2].

0.1 About this workshop

Causal mediation analysis can provide a mechanistic understanding of how an exposure impacts an outcome, a central goal in epidemiology and health sciences. However, rapid methodologic developments coupled with few formal courses presents challenges to implementation. Beginning with an overview of classical direct and indirect effects, this workshop will present recent advances that overcome limitations of previous methods, allowing for: (i) continuous exposures, (ii) multiple, non-independent mediators, and (iii) effects identifiable in the presence of intermediate confounders affected by exposure. Emphasis will be placed on flexible, stochastic and interventional direct and indirect effects, highlighting how these may be applied to answer substantive epidemiological questions from real-world studies. Multiply robust, nonparametric estimators of these causal effects, and free and open source R packages (medshift[3] and medoutcon[4]) for their application, will be introduced.

To ensure translation to real-world data analysis, this workshop will incorporate hands-on R programming exercises to allow participants practice in implementing the statistical tools presented. It is recommended that participants have working knowledge of the basic

[1] `https://epiresearch.org/annual-meeting/2021-meeting/workshop/`

[2] `https://code.nimahejazi.org/ser2021_mediation_workshop/ser2021mediation.pdf`

[3] `https://github.com/nhejazi/medshift`

[4] `https://github.com/nhejazi/medoutcon`

notions of causal inference, including counterfactuals and identification (linking the causal effect to a parameter estimable from the observed data distribution). Familiarity with the R programming language is also recommended.

0.2 Workshop schedule

- 10:00A-10:30A: introductions/mediation set up
- 10:30A-11:15A: controlled direct effects, natural direct and indirect effects, interventional direct and indirect effects
- 11:15A-11:45A: stochastic mediation estimands
- 11:45A-12:00P: how to choose an estimand in real-world examples
- 12:00P-12:15P break/discussion
- 12:15P-12:45P: what is the EIF
- 12:45P-01:00P: using the EIF for estimating the natural direct effect
- 01:00P-01:45P: practice: R code for estimation
- 01:45P-02:00P wrap up

NOTE: All times listed in Pacific Time.

0.3 About the instructors

Iván Díaz

I am an Assistant Professor at Weill Cornel Medicine. My research focuses on the development of non-parametric statistical methods for causal inference from observational and randomized studies with complex datasets, using machine learning. This includes but is not limited to mediation analysis, methods for continuous exposures, longitudinal data including survival analysis, and efficiency guarantees with covariate adjustment in randomized trials. I am also interested in general semi-parametric theory, machine learning, and high-dimensional data.

Nima Hejazi

I am a PhD candidate in biostatistics at UC Berkeley, working under the joint direction of Mark van der Laan and Alan Hubbard. My research interests fall at the intersection of causal inference and machine learning, drawing on ideas from non/semi-parametric estimation in large, flexible statistical models to develop efficient and robust statistical procedures for evaluating complex target estimands in observational and randomized studies. Particular areas of current emphasis include causal mediation/path analysis, outcome-dependent sampling designs, targeted loss-based estimation, and applications in vaccine efficacy trials. I am also passionate about statistical computing and open source software development for applied statistics.

Kara Rudolph

I am an Assistant Professor of Epidemiology at Columbia University. My research interests are in developing and applying causal inference methods to understand social and contextual influences on mental health, substance use, and violence in disadvantaged, urban areas of the United States. My current work focuses on developing methods for transportability and mediation, and subsequently applying those methods to understand how aspects of the school and peer environments mediate relationships between neighborhood factors and adolescent drug use across populations. More generally, my work on generalizing/transporting findings from study samples to target populations and identifying subpopulations most likely to benefit from interventions contributes to efforts to optimally target available policy and program resources.

0.4 Reproduciblity

These workshop materials were written using bookdown[5], and the complete source is available on GitHub[6]. This version of the book was built with R version 4.0.5 (2021-03-31), pandoc[7] version r rmarkdown::pandoc_version(), and the following packages:

0.5 Setup instructions

[5] http://bookdown.org/

[6] https://github.com/tlverse/tlverse-handbook

[7] https://pandoc.org/

0.5.1 R and RStudio

R and **RStudio** are separate downloads and installations. R is the underlying statistical computing environment. RStudio is a graphical integrated development environment (IDE) that makes using R much easier and more interactive. You need to install R before you install RStudio.

0.5.1.1 Windows

0.5.1.1.1 If you already have R and RStudio installed

- Open RStudio, and click on "Help" > "Check for updates". If a new version is available, quit RStudio, and download the latest version for RStudio.
- To check which version of R you are using, start RStudio and the first thing that appears in the console indicates the version of R you are running. Alternatively, you can type sessionInfo(), which will also display which version of R you are running. Go on the CRAN website[8] and check whether a more recent version is available. If so, please download and install it. You can check here[9] for more information on how to remove old versions from your system if you wish to do so.

0.5.1.1.2 If you don't have R and RStudio installed

- Download R from the CRAN website[10].
- Run the .exe file that was just downloaded
- Go to the RStudio download page[11]
- Under *Installers* select **RStudio x.yy.zzz - Windows XP/Vista/7/8** (where x, y, and z represent version numbers)
- Double click the file to install it
- Once it's installed, open RStudio to make sure it works and you don't get any error messages.

[8] https://cran.r-project.org/bin/windows/base/

[9] https://cran.r-project.org/bin/windows/base/rw-FAQ.html#How-do-I-UNinstall-R_003f

[10] http://cran.r-project.org/bin/windows/base/release.htm

[11] https://www.rstudio.com/products/rstudio/download/#download

0.5.1.2 macOS / Mac OS X

0.5.1.2.1 If you already have R and RStudio installed

- Open RStudio, and click on "Help" > "Check for updates". If a new version is available, quit RStudio, and download the latest version for RStudio.
- To check the version of R you are using, start RStudio and the first thing that appears on the terminal indicates the version of R you are running. Alternatively, you can type `sessionInfo()`, which will also display which version of R you are running. Go on the CRAN website[12] and check whether a more recent version is available. If so, please download and install it.

0.5.1.2.2 If you don't have R and RStudio installed

- Download R from the CRAN website[13].
- Select the `.pkg` file for the latest R version
- Double click on the downloaded file to install R
- It is also a good idea to install XQuartz[14] (needed by some packages)
- Go to the RStudio download page[15]
- Under *Installers* select **RStudio x.yy.zzz - Mac OS X 10.6+ (64-bit)** (where x, y, and z represent version numbers)
- Double click the file to install RStudio
- Once it's installed, open RStudio to make sure it works and you don't get any error messages.

0.5.1.3 Linux

- Follow the instructions for your distribution from CRAN[16], they provide information to get the most recent version of R for common distributions. For most distributions, you could use your package manager (e.g., for Debian/Ubuntu run `sudo apt-get install r-base`, and for Fedora `sudo yum install R`), but we don't recommend this approach as the versions provided by this are usually out of date. In any case, make sure you have at least R 3.3.1.

[12] https://cran.r-project.org/bin/macosx/

[13] http://cran.r-project.org/bin/macosx

[14] https://www.xquartz.org/

[15] https://www.rstudio.com/products/rstudio/download/#download

[16] https://cloud.r-project.org/bin/linux

- Go to the RStudio download page[17]
- Under *Installers* select the version that matches your distribution, and install it with your preferred method (e.g., with Debian/Ubuntu sudo dpkg −i rstudio −x.yy.zzz−amd64.deb at the terminal).
- Once it's installed, open RStudio to make sure it works and you don't get any error messages.

These setup instructions are adapted from those written for Data Carpentry: R for Data Analysis and Visualization of Ecological Data[18].

[17] https://www.rstudio.com/products/rstudio/download/#download

[18] http://www.datacarpentry.org/R-ecology-lesson/

Chapter 1

Preliminaries on causal mediation analysis

1.1 Motivating study

Do differences in the effects of treatment (comparing two medications for opioid use disorder, naltrexone vs buprenorphine) on risk of relapse operate through mediators of adherence, opioid use, pain, and depressive symptoms? (Rudolph et al., 2020)

1.2 What is causal mediation analysis?

- Causal mediation analyses assess how the paths behave under interventions
- Statistical mediation analyses assess associations between the variables

1.2.1 Why are the methods that we will discuss today important?

- Assume you are interested in the effect of a treatment assignment A (naltrexone vs. buprenorphine) on an outcome Y (risk of relapse) through mediators M (opioid use, pain, depressive symptoms)
- We have pre-treatment confounders W
- There is a confounder Z of $M \to Y$ ffected by treatment assignment (adherence)

- We could fit the following models:

$$\mathbb{E}(M \mid A = a, W = w, Z = z) = \gamma_0 + \gamma_1 a + \gamma_2 w + \gamma_3 z \tag{1.1}$$

$$\mathbb{E}(Y \mid M = m, A = a, W = w, Z = z) = \beta_0 + \beta_1 m + \beta_2 a + \beta_3 w + \beta_4 z \tag{1.2}$$

- The product $(\gamma_1\beta_1)$ has been proposed as a measure of the effect of A on Y through M
- Causal interpretation problems with this method:
  - We will see that this parameter cannot be interpreted as a causal effect

1.2.2 R Example:

- Assume we have a pre-treamtment confounder of Y and M, denote it with W
- For simplicity, assume A is randomized
- We'll generate a really large sample from a data generating mechanism so that we are not concerned with sampling errors

```
n <- 1e7
w <- rnorm(n)
a <- rbinom(n, 1, 0.5)
z <- rbinom(n, 1, 0.2 * a + 0.3)
m <- rnorm(n, w + z)
y <- rnorm(n, m + w - a + z)
```

- Note that the indirect effect (i.e., the effect through M) in this example is nonzero (there is a pathway $A \rightarrow Z \rightarrow M \rightarrow Y$)
- Let's see what the product of coefficients method would say:

```
lm_y <- lm(y ~ m + a + w + z)
lm_m <- lm(m ~ a + w + z)
## product of coefficients
coef(lm_y)[2] * coef(lm_m)[2]
```

Among other things, in this workshop:

- We will provide some understanding for why the above method fails in this example
- We will study estimators that are robust to misspecification in the above models

1.3 Causal mediation models

In this workshop we will use directed acyclic graphs. We will focus on the two types of graph:

1.3.1 No intermediate confounders

1.3.2 Intermediate confounders

The above graphs can be interpreted as a *non-parametric structural equation model* (NPSEM), also known as *structural causal model* (SCM):

$$\begin{aligned} W &= f_W(U_W) && (1.3)\\ A &= f_A(W, U_A) && (1.4)\\ Z &= f_Z(W, A, U_Z) && (1.5)\\ M &= f_M(W, A, Z, U_M) && (1.6)\\ Y &= f_Y(W, A, Z, M, U_Y) && (1.7) \end{aligned}$$

- Here $U = (U_W, U_A, U_Z, U_M, U_Y)$ is a vector of all unmeasured exogenous factors affecting the system
- The functions f are assumed fixed but unknown
- We posit this model as a system of equations that nature uses to geenrate the data
- Therefore we leave the functions f unspecified (i.e., we do not know the true nature mechanisms)
- Sometimes we know something: e.g., if A is randomized we know $A = f_A(U_A)$ where U_A is the flip of a coin (i.e., independent of everything).

1.4 Counterfactuals

- We define all the effects of interest using *counterfactuals*
- Counterfactuals are hypothetical random variables that would have been observed in an alternative world where something had happened, possibly contrary to fact
- Y_a is a counterfactual variable in a hypothetical world where $\mathbb{P}(A = a) = 1$ with probability one

- $Y_{a,m}$ is the counterfactual outcome in a world where $\mathbb{P}(A = a, M = m) = 1$
- M_a is the counterfactual variable representing the mediator in a world where $\mathbb{P}(A = a) = 1$.

1.4.1 How are counterfactuals defined?

- In the NPSEM framework, counterfactuals are quantities *derived* from the model.
- Take as example the DAG in Figure 1.2:

$$Y_a = f_Y(W, a, Z_a, M_a, U_Y) \tag{1.8}$$

$$Y_{a,m} = f_Y(W, a, Z_a, m, U_Y) \tag{1.9}$$

$$M_a = f_M(W, a, Z_a, U_M) \tag{1.10}$$

- You can also define *nested counterfactuals*
- For example, if A is binary, you can think of the following counterfactual

$$Y_{1,M_0} = f_Y(W, 1, Z_1, M_0, U_Y)$$

- Interpreted as *the outcome for an individual in a hypothetical world where treatment was given but the mediator was held at the value it would have taken under no treatment*
- Causal effects are defined in terms of the distribution of these counterfactuals.
- That is, causal effects give you information about what would have happened *in some hypothetical world*.

Chapter 2

Path-specific casual mediation effect types

- Controlled direct effects
- Natural direct and indirect effects
- Interventional direct and indirect effects

2.1 Controlled direct effects

- Set the mediator to a reference value $M = m$ uniformly for everyone in the population
- Compare $A = 1$ vs $A = 0$ with $M = m$ fixed

$$\psi_{\text{CDE}} = \mathbb{E}(Y_{1,m} - Y_{0,m})$$

2.1.1 Identification assumptions:

- Confounder assumptions:
    - $A \perp\!\!\!\perp Y_{a,m} \mid W$
    - $M \perp\!\!\!\perp Y_{a,m} \mid W, A$
- Positivity assumptions:

- $\mathbb{P}(M = m \mid A = a, W) > 0 \; a.e.$
- $\mathbb{P}(A = a \mid W) > 0 \; a.e.$

Under the above identification assumptions, the controlled direct effect can be identified:

$$\mathbb{E}(Y_{1,m} - Y_{0,m}) = \mathbb{E}\{\mathbb{E}(Y \mid A = 1, M = m, W) - \mathbb{E}(Y \mid A = 0, M = m, W)\}$$

- For intuition about this formula in R, let's continue with a toy example:

```
n <- 1e6
w <- rnorm(n)
a <- rbinom(n, 1, 0.5)
m <- rnorm(n, w + a)
y <- rnorm(n, w + a + m)
```

- First we fit a correct model for the outcome

```
lm_y <- lm(y ~ m + a + w)
```

- Assume we would like the CDE at $m = 0$
- Then we generate predictions $\mathbb{E}(Y \mid A = 1, M = m, W)$ and $\mathbb{E}(Y \mid A = 0, M = m, W)$:

```
pred_y1 <- predict(lm_y, newdata = data.frame(a = 1, m = 0, w = w))
pred_y0 <- predict(lm_y, newdata = data.frame(a = 0, m = 0, w = w))
```

- Then we compute the difference between the predicted values $\mathbb{E}(Y \mid A = 1, M = m, W) - \mathbb{E}(Y \mid$ and average across values of W

```
## CDE at m = 0
mean(pred_y1 - pred_y0)
```

2.1.2 Is this the estimand I want?

- Makes the most sense if can intervene directly on M
    - And can think of a policy that would set everyone to a single constant level $m \in \mathcal{M}$.
    - J. Pearl calls this *prescriptive*.
    - Can you think of an example?
    - Air pollution, rescue inhaler dosage, hospital visits
    - Does not provide a decomposition of the average treatment effect into direct and indirect effects

What if our research question doesn't involve intervening directly on the mediator?

What if we want to decompose the average treatment effect into its direct and indirect counterparts?

2.2 Natural direct and indirect effects

Still using the same DAG as above,

- Recall the definition of the nested counterfactual

$$Y_{1,M_0} = f_Y(W, 1, Z_1, M_0, U_Y)$$

- Interpreted as *the outcome for an individual in a hypothetical world where treatment was given but the mediator was held at the value it would have taken under no treatment*

- Recall that, because of the definition of counterfactuals

$$Y_{1,M_1} = Y_1$$

Then we can decompose the *average treatment effect* $E(Y_1 - Y_0)$ as follows

$$\mathbb{E}[Y_{1,M_1} - Y_{0,M_0}] = \underbrace{\mathbb{E}[Y_{1,M_1} - Y_{1,M_0}]}_{\text{natural indirect effect}} + \underbrace{\mathbb{E}[Y_{1,M_0} - Y_{0,M_0}]}_{\text{natural direct effect}}$$

- Natural direct effect (NDE): Varying treatment while keeping the mediator fixed at the value it would have taken under no treatment
- Natural indirect effect (NIE): Varying the mediator from the value it would have taken under treatment to the value it would have taken under control, while keeping treatment fixed

2.2.1 Identification assumptions:

- $A \perp\!\!\!\perp Y_{a,m} \mid W$
- $M \perp\!\!\!\perp Y_{a,m} \mid W, A$
- $A \perp\!\!\!\perp M_a \mid W$
- $M_0 \perp\!\!\!\perp Y_{1,m} \mid W$
- and positivity assumptions

2.2.2 Cross-world independence assumption

What does $M_0 \perp\!\!\!\perp Y_{1,m} \mid W$ mean?

- Conditional on W, knowledge of the mediator value in the absence of treatment, M_0, provides no information about the outcome under treatment, $Y_{1,m}$.
- Can you think of a data-generating mechanism that would violate this assumption?
- Example: in a randomized study, whenever we believe that treatment assignment works through adherence (i.e., almost always), we are violating this assumption (more on this later).
- Cross-world assumptions are problematic for other reasons, including:
    - You can never design a randomized study where the assumption holds by design.

If the cross-world assumption holds, can write the NDE as a weighted average of controlled direct effects at each level of $M = m$.

$$\mathbb{E} \sum_m \{\mathbb{E}(Y_{1,m} \mid W) - \mathbb{E}(Y_{0,m} \mid W)\} \mathbb{P}(M_0 = m \mid W)$$

- If CDE(m) is constant across m, then CDE = NDE.

2.2.3 Identification formula:

- Under the above identification assumptions, the natural direct effect can be identified:
  $\mathbb{E}(Y_{1,M_0} - Y_{0,M_0}) = \mathbb{E}[\mathbb{E}\{\mathbb{E}(Y \mid A = 1, M, W) - \mathbb{E}(Y \mid A = 0, M, W) | A = 0, W\}]$

- The natural indirect effect can be identified similarly.
- Let's dissect this formula in R:

```
n <- 1e6
w <- rnorm(n)
a <- rbinom(n, 1, 0.5)
m <- rnorm(n, w + a)
y <- rnorm(n, w + a + m)
```

- First we fit a correct model for the outcome

```
lm_y <- lm(y ~ m + a + w)
```

- Then we generate predictions
  $\mathbb{E}(Y \mid A = 1, M, W)$ and $\mathbb{E}(Y \mid A = 0, M, W)$

  with A fixed but letting M and W take their observed values

```
pred_y1 <- predict(lm_y, newdata = data.frame(a = 1, m = m, w = w))
pred_y0 <- predict(lm_y, newdata = data.frame(a = 0, m = m, w = w))
```

- Then we compute the difference between the predicted values
  $\mathbb{E}(Y \mid A = 1, M, W) - \mathbb{E}(Y \mid A = 0, M, W),$

- and use this difference as a pseudo-outcome in a regression on A and W:
  $\mathbb{E}\{\mathbb{E}(Y \mid A = 1, M, W) - \mathbb{E}(Y \mid A = 0, M, W) | A = 0, W\}$

```
pseudo <- pred_y1 - pred_y0
lm_pseudo <- lm(pseudo ~ a + w)
```

- Now we predict the value of this pseudo-outcome under $A = 0$, and average the result

```
pred_pseudo <- predict(lm_pseudo, newdata = data.frame(a = 0, w = w))
## NDE:
mean(pred_pseudo)
```

2.2.4 Is this the estimand I want?

- Makes sense to intervene on A but not directly on M.
- Want to understand a natural mechanism underlying an association/ total effect. J. Pearl calls this *descriptive*.
- NDE + NIE = total effect (ATE).
- Okay with the assumptions.

What if our data structure involves a post-treatment confounder of the mediator-outcome relationship (e.g., adherence)?

2.2.5 Unidentifiability of the NDE and NIE in this setting

- In this example, natural direct and indirect effects are unidentifiable from observed data $O = (W, A, Z, M, Y)$.
- The reason for this is that the cross-world counterfactual assumption
$$Y_{1,m} \perp\!\!\!\perp M_0 \mid W$$
does not hold in the above directed acyclic graph.
- Technically, the reason for this is that an intervention setting $A = 1$ (necessary for the definition of $Y_{1,m}$) induces a counterfactual variable Z_1.
- Likewise, an intervention setting $A = 0$ (necessary for the definition of M_0) induces a counterfactual Z_0.
- The variables Z_1 and Z_0 are correlated because they share unmeasured common causes, U_Z.

- The variable Z_1 is correlated with $Y_{1,m}$, and the variable Z_0 is correlated with M_0, because they are counterfactual outcomes in the same hypothetical worlds.
- To see this in the definition of counterfactual from a causal structural model:

$$Y_{1,m} = f_Y(W, 1, Z_1, m, U_Y), \text{ and}$$
$$M_0 = f_M(W, 0, Z_0, U_M)$$

are correlated even after adjusting for W by virtue of Z_1 and Z_0 being correlated.

Intuitively:

- Z is a confounder of the relation $M \to Y$, which requires adjustment
- But Z is on the pathway $A \to Y$, which precludes adjustment

Note: CDEs are still identified in this setting. They can be identified and estimated similarly to a longitudinal data sructure with a two-time-point intervention.

2.3 Interventional (in)direct effects

- Let G_a denote a random draw from the distribution of $M_a \mid W$
- Define the counterfactual Y_{1,G_0} as the counterfactual variable in a hypothetical world where A is set $A = 1$ and M is set to $M = G_0$ with probability one.
- Define Y_{0,G_0} and Y_{1,G_1} similarly
- Then we can define:

$$\mathbb{E}[Y_{1,G_1} - Y_{0,G_0}] = \underbrace{\mathbb{E}[Y_{1,G_1} - Y_{1,G_0}]}_{\text{interventional indirect effect}} + \underbrace{\mathbb{E}[Y_{1,G_0} - Y_{0,G_0}]}_{\text{interventional direct effect}}$$

- Note that $\mathbb{E}[Y_{1,G_1} - Y_{0,G_0}]$ is still a *total effect* of treatment, even if it is different from the ATE $\mathbb{E}[Y_1 - Y_0]$
- We gain in the ability to solve a problem, but lose in terms of interpretation of the causal effect (cannot decompose the ATE)

2.3.1 An alternative definition of the effects:

- Above we defined G_a as a random draw from the distribution of $M_a \mid W$

- What if instead we define G_a as a random draw from the distribution of $M_a \mid (Z_a, W)$
- It turns out the indirect effect defined in this way only measures the path $A \to M \to Y$, and not the path $A \to Z \to M \to Y$
- There may be important reasons to choose one over another (e.g., survival analyses where we want the distribution conditional on Z, instrumental variable designs where it doesn't make sense to condition on Z)

2.3.2 Identification assumptions:

- $A \perp\!\!\!\perp Y_{a,m} \mid W$
- $M \perp\!\!\!\perp Y_{a,m} \mid W, A, Z$
- $A \perp\!\!\!\perp M_a \mid W$
- and positivity assumptions.

Under these assumptions, the population interventional direct and indirect effect is identified:

$$\mathbb{E}(Y_{a,G_{a'}}) =$$

$$\mathbb{E}\left[\mathbb{E}\left\{\sum_z \mathbb{E}(Y \mid A = a, Z = z, M, W)\mathbb{P}(Z = z \mid A = a, W) \mid A = a', W\right\}\right]$$

- Let's dissect this formula in R:

```
n <- 1e6
w <- rnorm(n)
a <- rbinom(n, 1, 0.5)
z <- rbinom(n, 1, 0.5 + 0.2 * a)
m <- rnorm(n, w + a - z)
y <- rnorm(n, w + a + z + m)
```

- Let us compute $\mathbb{E}(Y_{1,G_0})$ (so that $a = 1$, and $a' = 0$).
- First, fit a regression model for the outcome, and compute
  $$\mathbb{E}(Y \mid A = a, Z = z, M, W)$$
  for all values of z

```r
lm_y <- lm(y ~ m + a + z + w)
pred_a1z0 <- predict(lm_y, newdata = data.frame(m = m, a = 1, z = 0, w = w)
pred_a1z1 <- predict(lm_y, newdata = data.frame(m = m, a = 1, z = 1, w = w)
```

- Now we fit the true model for $Z \mid A, W$ and get the conditional probability that $Z = 1$ fixing $A = 1$

```r
prob_z <- lm(z ~ a)
pred_z <- predict(prob_z, newdata = data.frame(a = 1))
```

- Now we compute the following pseudo-outcome:

$$\sum_z \mathbb{E}(Y \mid A = a, Z = z, M, W)\mathbb{P}(Z = z \mid A = a, w)$$

```r
pseudo_out <- pred_a1z0 * (1 - pred_z) + pred_a1z1 * pred_z
```

- Now we regress this pseudo-outcome on A, W, and compute the predictions setting $A = 0$, that is,

$$\mathbb{E}\left\{\sum_z \mathbb{E}(Y \mid A = a, Z = z, M, W)\mathbb{P}(Z = z \mid A = a, w) \mid A = a', W\right\}$$

```r
fit_pseudo <- lm(pseudo_out ~ a + w)
pred_pseudo <- predict(fit_pseudo, data.frame(a = 0, w = w))
```

- And finally, just average those predictions!

```r
## Mean(Y(1, G(0)))
mean(pred_pseudo)
```

- This was for $(a, a') = (1, 0)$. Can do the same with $(a, a') = (1, 1)$, and $(a, a') = (0, 0)$ to obtain an effect decomposition

$$\mathbb{E}[Y_{1,G_1} - Y_{0,G_0}] = \underbrace{\mathbb{E}[Y_{1,G_1} - Y_{1,G_0}]}_{\text{interventional indirect effect}} + \underbrace{\mathbb{E}[Y_{1,G_0} - Y_{0,G_0}]}_{\text{interventional direct effect}}$$

2.3.3 Is this the estimand I want?

- Makes sense to intervene on A but not directly on M.
- Goal is to understand a natural mechanism underlying an association or total effect.
- Okay with the assumptions!

2.4 Estimand Summary

Chapter 3

Stochastic Direct and Indirect Effects

3.1 Definition of the effects

Consider the following directed acyclic graph.

3.2 Motivation for stochastic interventions

- So far we have discussed controlled, natural, and interventional (in)direct effects
- These effects require that $0 < \mathbb{P}(A = 1 \mid W) < 1$
- They are defined only for binary exposures
- *What can we do when the positivity assumption does not hold or the exposure is continuous?*
- Solution: we can use stochastic effects

3.3 Definition of stochastic effects

There are two possible ways of defining stochastic effects:

- Consider the effect of an intervention where the exposure is drawn from a distribution
    - For example incremental propensity score interventions

- Consider the effect of an intervention where the post-intervention exposure is a function of the actually received exposure
  - For example modified treatment policies
- In both cases $A \mid W$ is a non-deterministic intervention, thus the name *stochastic intervention*

Example: incremental propensity score interventions (IPSI) (Kennedy, 2018)

Definition of the intervention

- Assume A is binary, and $\mathbb{P}(A = 1 \mid W = w) = g(1 \mid w)$ is the propensity score
- Consider an intervention in which each individual receives the intervention with probability $g_\delta(1 \mid w)$, equal to
$$g_\delta(1 \mid w) = \frac{\delta g(1 \mid w)}{\delta g(1 \mid w) + 1 - g(1 \mid w)}$$
- e.g., draw the post-intervention exposure from a Bernoulli variable with probability $g_\delta(1 \mid w)$
- The value δ is user given
- Let A_δ denote the post-intervention exposure distribution
- Some algebra shows that δ is an odds ratio comparing the pre- and post-intervention exposure distributions
$$\delta = \frac{\text{odds}(A_\delta = 1 \mid W = w)}{\text{odds}(A = 1 \mid W = w)}$$
- Interpretation: *what would happen in a world where the odds of receiving treatment is increased by δ*
- Let Y_{A_δ} denote the outcome in this hypothetical world

3.3.0.1 Illustrative application for IPSIs

- Consider the effect of participation in sports on children's BMI
- Mediation through snacking, exercising, etc.
- Intervention: for each individual, increase the odds of participating in sports by $\delta = 2$
- The post-intervention exposure is a draw A_δ from a Bernoulli distribution with probability $g_\delta(1 \mid w)$

Example: modified treatment policies (MTP) (Díaz and Hejazi, 2020)

Definition of the intervention

- Consider a continuous exposure A taking values in the real numbers
- Consider an intervention that assigns exposure as $A_\delta = A - \delta$
- Example: A is pollution measured as $PM_{2.5}$ and you are interested in an intervention that reduces $PM_{2.5}$ concentration by some amount δ

3.3.1 Mediation analysis for stochastic interventions

- The total effect of an IPSI can be computed as a contrast of the outcome under intervention vs no intervention:
$$\psi = \mathbb{E}[Y_{A_\delta} - Y]$$

- Recall the NPSEM
$$\begin{aligned} W &= f_W(U_W) && (3.1)\\ A &= f_A(W, U_A) && (3.2)\\ M &= f_M(W, A, U_M) && (3.3)\\ Y &= f_Y(W, A, M, U_Y) && (3.4)\end{aligned}$$

- From this we have
$$\begin{aligned} M_{A_\delta} &= f_M(W, A_\delta, U_M)\\ Y_{A_\delta} &= f_Y(W, A_\delta, M_{A_\delta}, U_Y)\end{aligned}$$

- Thus, we have $Y_{A_\delta} = Y_{A_\delta, M_{A_\delta}}$ and $Y = Y_{A, M_A}$

- Let us introduce the counterfactual $Y_{A_\delta, M}$, interpreted as the outcome observed in a world where the intervention on A is performed but the mediator is fixed at the value it would have taken under no intervention:
$$Y_{A_\delta, M} = f_Y(W, A_\delta, M_{A_\delta}, U_Y)$$

- Then we can decompose the total effect into:
$$\mathbb{E}[Y_{A_\delta, M_{A_\delta}} - Y_{A, M_A}] = \underbrace{\mathbb{E}[Y_{A_\delta, M_{A_\delta}} - Y_{A_\delta, M}]}_{\text{stochastic natural indirect effect}} + \underbrace{\mathbb{E}[Y_{A_\delta, M} - Y_{A, M}]}_{\text{stochastic natural direct effect}}$$

3.4 Identification assumptions

- Confounder assumptions:
  - $A \perp\!\!\!\perp Y_{a,m} \mid W$
  - $M \perp\!\!\!\perp Y_{a,m} \mid W, A$
- No confounder of $M \to Y$ affected by A
- Positivity assumptions:
  - If $g_\delta(a \mid w) > 0$ then $g(a \mid w) > 0$
  - If $\mathbb{P}(Z = z \mid W = w) > 0$ then $\mathbb{P}(Z = z \mid A = a, W = w) > 0$

Under these assumptions, stochastic effects are identified as follows

- The indirect effect can be identified as follows

$$\mathbb{E}(Y_{A_\delta} - Y_{A_\delta,M}) =$$

$$\mathbb{E}\left[\sum_a \{\mathbb{E}(Y \mid A = a, W) - \mathbb{E}(Y \mid A = a, M, W)\} g_\delta(a \mid W)\right]$$

- The direct effect can be identified as follows

$$\mathbb{E}(Y_{A_\delta} - Y_{A_\delta,M}) =$$

$$\mathbb{E}\left[\sum_a \{\mathbb{E}(Y \mid A = a, M, W) - Y\} g_\delta(a \mid W)\right]$$

- Let's dissect the formula for the indirect effect in R:

```
n <- 1e6
w <- rnorm(n)
a <- rbinom(n, 1, plogis(1 + w))
m <- rnorm(n, w + a)
y <- rnorm(n, w + a + m)
```

- First, fit regressions of the outcome on (A, W) and (M, A, W):

```
fit_y1 <- lm(y ~ m + a + w)
fit_y2 <- lm(y ~ a + w)
```

- Get predictions fixing $A = a$ for all possible values a

```
pred_y1_a1 <- predict(fit_y1, newdata = data.frame(a = 1, m, w))
pred_y1_a0 <- predict(fit_y1, newdata = data.frame(a = 0, m, w))
pred_y2_a1 <- predict(fit_y2, newdata = data.frame(a = 1, w))
pred_y2_a0 <- predict(fit_y2, newdata = data.frame(a = 0, w))
```

- Compute
$$\{\mathbb{E}(Y \mid A = a, W) - \mathbb{E}(Y \mid A = a, M, W)\}$$
for each value a

```
pseudo_a1 <- pred_y2_a1 - pred_y1_a1
pseudo_a0 <- pred_y2_a0 - pred_y1_a0
```

- Estimate the propensity score $g(1 \mid w)$ and evaluate the post-intervention propensity score $g_\delta(1 \mid w)$

```
pscore_fit <- glm(a ~ w, family = binomial())
pscore <- predict(pscore_fit, type = 'response')
## How do the intervention vs observed propensity score compare
pscore_delta <- 2 * pscore / (2 * pscore + 1 - pscore)
```

- What do the post-intervention propensity scores look like?

```
plot(pscore, pscore_delta, xlab = 'Observed␣prop.␣score',
     ylab = 'Prop.␣score␣under␣intervention')
abline(0, 1)
```

3.5 What are the odds of exposure under intervention vs real world?

```
odds <- (pscore_delta / (1 - pscore_delta)) / (pscore / (1 - pscore))
summary(odds)
```

- Compute the sum

$$\sum_a \{\mathbb{E}(Y \mid A = a, W) - \mathbb{E}(Y \mid A = a, M, W)\} g_\delta(a \mid W)$$

```
indirect <- pseudo_a1 * pscore_delta + pseudo_a0 * (1 - pscore_delta)
```

- The average of this value is the indirect effect

```
## E[Y(Adelta) - Y(Adelta, M)]
mean(indirect)
```

- The direct effect is

$$\mathbb{E}(Y_{A_\delta} - Y_{A_\delta,M}) = \mathbb{E}\left[\sum_a \{\mathbb{E}(Y \mid A = a, M, W) - Y\} g_\delta(a \mid W)\right]$$

- Which can be computed as

```
direct <- (pred_y1_a1 - y) * pscore_delta +
          (pred_y1_a0 - y) * (1 - pscore_delta)
mean(direct)
```

3.6 Summary

- Stochastic (in)direct effects
  - Relax the positivity assumption
  - Can be defined for non-binary exposures
  - Do not require a cross-world assumption
- Still require the absence of intermediate confounders
  - But, compared to the NDE and NIE, we can design a randomized study where identifiability assumptions hold, at least in principle
  - There is a version of these effects that can accomodate intermediate confounders (Hejazi et al., 2020)
  - R implementation to be released soon, stay tuned!

Chapter 4

How to choose an estimand: Real world example

4.1 Comparative effectivness of two medications for opioid use disorder (OUD)

Motivation: Opposite overall treatment effects for homeless versus nonhomeless participants.

4.1.1 Getting specific about the question

To what extent does the indirect effect through mediators of adherence, pain, and depressive symptoms explain the differences in treatment effects on OUD relapse for homeless and nonhomeless individuals?

What estimand do we want?

- Can we set $M = m$ (i.e., same value) for everyone?
- Are we interested in estimating indirect effects?

$\rightarrow$ So, *not* controlled direct effect.

- Do we have an intermediate confounder?

- Yes, and it's important.

$\rightarrow$ So, *not* natural (in)direct effects.

- So, we're left with the interventional direct and indirect effects.
- Do we want to estimate the path through treatment initiation (Z)?
- Yes, so, *not* the conditional versions of these effects.
- Estimands:
  - Direct effect: $\mathbb{E}(Y_{1,G_0} - Y_{0,G_0})$
  - Indirect effect: $\mathbb{E}(Y_{1,G_1} - Y_{1,G_0})$
- Here G_a is a draw from the distribution of $M_a \mid W$.
- Need to incorporate multiple and continuous mediators

What if the positivity assumption $\mathbb{P}(A = a \mid W) > 0$ violated?

$\rightarrow$ Can't identify or estimate any of the above effects

- But we can estimate the effect of some stochastic interventions, e.g., IPSIs
- Tradeoff between feasibility and interpretation

What if the exposure variable is continuous?

$\rightarrow$ All the above effects are defined for binary exposures

- But we can estimate the effect of some stochastic interventions
- Work in progress (including upcoming R software)

Chapter 5

Preliminaries on semiparametric estimation

5.1 From causal to statistical quantities

- We have arrived at identification formulas that express quantities that we care about in terms of observable quantities
- This required **causal assumptions**
  - Many of these assumptions are empirically unverifiable
  - We saw an example where we could relax the cross-world assumption, at the cost of changing the parameter interpretation
  - and where we could relax the positivity assumption, also at the cost of changing the parameter interpretation
- The resulting estimation problem can be tackled using **statistical assumptions** of various degrees of strength
  - Most of these assumptions are verifiable (e.g., a linear model)
  - Thus, most are unnecessary (except for convenience)
  - The estimation approach we use reduces reliance on these statistical assumptions

5.1.1 Computing identification formulas if you know the true distribution

- The mediation parameters that we consider can be seen as a function of the joint probability distribution of $O = (W, A, Z, M, Y)$
- For example, under identifiability assumptions the natural direct effect is equal to $$\psi(\mathbb{P}) = \mathbb{E}[\mathbb{E}\{\mathbb{E}(Y \mid A = 1, M, W) - \mathbb{E}(Y \mid A = 0, M, W) \mid A = 0, W\}]$$
- The notation $\psi(\mathbb{P})$ implies that the parameter is a function of $\mathbb{P}$
- This means that we can compute it for any distribution $\mathbb{P}$
- For example, if we know the true $\mathbb{P}(W, A, M, Y)$, we can comnpute the true value of the parameter by:
  - Computing the conditional expectation $\mathbb{E}(Y \mid A = 1, M = m, W = w)$ for all values (m, w)
  - Computing the probability $\mathbb{P}(M = m \mid A = 0, W = w)$ for all values (m, w)
  - Computing the probability $\mathbb{P}(W = w)$ for all values w
  - Computing the mean over all values (m, w)

5.1.2 Estimating identification formulas

The above is how you would compute the *true value* **if you know** the true distribution $\mathbb{P}$

- This is exactly what we did in our R examples before
- But we can use the same logic for estimation:
  - Fit a regression to estimate, say $\hat{\mathbb{E}}(Y \mid A = 1, M = m, W = w)$
  - Fit a regression to estimate, say $\hat{\mathbb{P}}(M = m \mid A = 0, W = w)$
  - Estimate $\mathbb{P}(W = w)$ with the empirical distribution
  - Evaluate $$\psi(\hat{\mathbb{P}}) = \hat{\mathbb{E}}[\hat{\mathbb{E}}\{\hat{\mathbb{E}}(Y \mid A = 1, M, W) - \hat{\mathbb{E}}(Y \mid A = 0, M, W) \mid A = 0, W\}]$$
- This is known as the g-computation estimator

5.1.3 How can g-estimation be implemented in practice?

- There are two possible ways to do g-computation estimation:
  - Using parametric models for the above regressions
  - Using flexible data-adaptive regression (aka machine learning)

5.1.4 Pros and cons of parametric models

- Pros:
  - Easy to understand
  - Ease of implementation (standard regression software)
  - Can use the Delta method or the bootstrap for computation of standard errors
- Cons:
  - Unless W and M contain very few categorical variables, it is very easy to misspecify the models
  - This can introduce sizable bias in the estimators

5.1.5 An example of the bias of a g-computation estimator of the natural direct effect

- The following R chunk provides simulation code to exemplify the bias of a g-computation estimator in a simple situation

```r
mean_y <- function(m, a, w) abs(w) + a * m
mean_m <- function(a, w) plogis(w^2 - a)
pscore <- function(w) plogis(1 - abs(w))
```

- This yields a true NDE value of

```r
w_big <- runif(1e6, -1, 1)
trueval <- mean((mean_y(1, 1, w_big) - mean_y(1, 0, w_big)) *
  mean_m(0, w_big) + (mean_y(0, 1, w_big) - mean_y(0, 0, w_big)) *
    (1 - mean_m(0, w_big)))
print(trueval)
```

- Let's perform a simulation where we draw 1000 datasets from the above distribution, and compute a g-computation estimator based on

```r
gcomp <- function(y, m, a, w) {
  lm_y <- lm(y ~ m + a + w)
  pred_y1 <- predict(lm_y, newdata = data.frame(a = 1, m = m, w = w))
  pred_y0 <- predict(lm_y, newdata = data.frame(a = 0, m = m, w = w))
  pseudo <- pred_y1 - pred_y0
  lm_pseudo <- lm(pseudo ~ a + w)
  pred_pseudo <- predict(lm_pseudo, newdata = data.frame(a = 0, w = w)
  estimate <- mean(pred_pseudo)
  return(estimate)
}

estimate <- numeric(1000)
for (i in 1:1000) {
  n <- 1000
  w <- runif(n, -1, 1)
  a <- rbinom(n, 1, pscore(w))
  m <- rbinom(n, 1, mean_m(a, w))
  y <- rnorm(n, mean_y(m, a, w))

  estimate[i] <- gcomp(y, m, a, w)
}

hist(estimate)
abline(v = trueval, col = "red", lwd = 4)
```

- The bias also affects the confidence intervals:

```r
cis <- cbind(
  estimate - qnorm(0.975) * sd(estimate),
  estimate + qnorm(0.975) * sd(estimate)
)

ord <- order(rowSums(cis))
lower <- cis[ord, 1]
upper <- cis[ord, 2]
curve(trueval + 0 * x,
  ylim = c(0, 1), xlim = c(0, 1001), lwd = 2, lty = 3, xaxt = "n",
  xlab = "", ylab = "Confidence interval", cex.axis = 1.2, cex.lab = 1
)
for (i in 1:1000) {
  clr <- rgb(0.5, 0, 0.75, 0.5)
  if (upper[i] < trueval || lower[i] > trueval) clr <- rgb(1, 0, 0, 1)
  points(rep(i, 2), c(lower[i], upper[i]), type = ”l”, lty = 1, col = clr)
}
text(450, 0.10, ”n=1000␣repetitions␣=␣1000␣”, cex = 1.2)
text(450, 0.01, paste0(
  ”Coverage␣probability␣=␣”,
  mean(lower < trueval & trueval < upper), ”%”
), cex = 1.2)
```

5.1.6 Pros and cons of g-computation with data-adaptive regression

- Pros:
  - Easy to understand
  - Alleviate model-misspecification bias
- Cons:
  - Might be harder to implement depending on the regression procedures used
  - No general approaches for computation of standard errors and confidence intervals
  - For example, the bootstrap is not guaranteed to work, and it is known to fail in some cases

5.2 Semiparametric estimation (or correcting the bias of g-computation estimators)

- G-computation estimation with data-adaptive regression offers an incorrect bias/-variance trade-off
- It accepts more bias than necessary
- The bias of a g-computation estimator may be corrected as follows:

$$\psi(\hat{\mathbb{P}}) + \frac{1}{n}\sum_{i=1}^{n} D(O_i)$$

  for some function $D(O_i)$ of the data

- The function $D(O)$ is called *the efficient influence function* (EIF)
- The EIF must be found on a case-by-case basis for each parameter $\psi(\mathbb{P})$
- For example, for estimating the standardized mean $\psi(\mathbb{P}) = \mathbb{E}[\mathbb{E}(Y \mid A = 1, W)]$, we have
$$D(O) = \frac{A}{\hat{\mathbb{P}}(A = 1 \mid W)}[Y - \hat{\mathbb{E}}(Y \mid A = 1, W)] + \hat{\mathbb{E}}(Y \mid A = 1, W) - \psi(\hat{\mathbb{P}})$$
- The EIF is found by using a distributional analogue of a Taylor expansion
- In this workshop we will omit the specific form of $D(O)$ for some of the parameters that we use
- But the estimators we discuss and implement in the R packages will be based on these EIFs
- And the specific form of the EIF may be found in papers in the references

Note: the bias correction above may have an additional problem of returning parameter estimates outside of natural bounds. E.g., probabilities greater than one. A solution to this (not discussed in this workshop) is targeted minimum loss based estimation.

Chapter 6

Using the EIF to construct an estimator: the case of the natural direct effect

6.1 Natural direct effect

Recall:

- Assuming a binary A, we define the natural direct effect as:
$$NDE = E(Y_{1,M_0} - Y_{0,M_0}),$$
- and the natural indirect effect as:
$$NIE = E(Y_{1,M_1} - Y_{1,M_0}).$$
- The observed data is $O = (W, A, M, Y)$

This SCM is represented in the above DAG and the following causal models:

$$\begin{aligned} W &= f_W(U_W) \\ A &= f_A(W, U_A) \\ M &= f_M(W, A, U_M) \\ Y &= f_Y(W, A, M, U_Y), \end{aligned}$$

where (U_W, U_A, U_M, U_Y) are exogenous random errors.

We assume - A is a single binary randomized treatment (and thus $A = f_A(U_A)$) - M is a single binary mediator - There are no restrictions on the distribution of W or Y

Recall that we need to assume the following to identify the above caual effects from our observed data:

- $A \perp\!\!\!\perp Y_{a,m} \mid W$
- $M \perp\!\!\!\perp Y_{a,m} \mid W, A$
- $A \perp\!\!\!\perp M_a \mid W$
- $M_0 \perp\!\!\!\perp Y_{1,m} \mid W$
- and positivity assumptions

Then, the NDE is identified as

$$\psi(\mathbb{P}) = \mathbb{E}[\mathbb{E}\{\mathbb{E}(Y \mid A = 1, M, W) - \mathbb{E}(Y \mid A = 0, M, W) \mid A = 0, W\}]$$

6.1.1 The efficient influence function for the NDE

- For illustration, we will first present how to construct an estimator of the NDE that uses the EIF "by hand"
- For other parameters, we will teach you how to use our packages *medoutcon* and *medshift*

First, we need to introduce some notation to describe the EIF for the NDE

- Let $Q(M, W)$ denote $\mathbb{E}(Y \mid A = 1, M, W) - \mathbb{E}(Y \mid A = 0, M, W)$
- We can now introduce the EIF:
$$\begin{aligned} D(O) = &\left\{ \frac{I(A=1)}{\mathbb{P}(A = 1 \mid W)} \frac{\mathbb{P}(M \mid A = 0, W)}{\mathbb{P}(M \mid A = 1, W)} - \frac{I(A=0)}{\mathbb{P}(A = 0 \mid W)} \right\} \times [Y - \mathbb{E}(Y \mid A, M, W)] \\ &+ \frac{I(A=0)}{\mathbb{P}(A = 0 \mid W)} \big\{ Q(M, W) - \mathbb{E}[Q(M, W) | W, A = 0] \big\} \\ &+ \mathbb{E}[Q(M, W) | W, A = 0] - \psi(\mathbb{P}) \end{aligned}$$
- Estimating $\mathbb{P}(M \mid A, W)$ is a really hard problem when M is high-dimensional. But, since we have the ratio of these conditional densitities, we can reparamterize using Bayes rule to get something that is easier to compute:

$$\frac{\mathbb{P}(M \mid A = 0, W)}{\mathbb{P}(M \mid A = 1, W)} = \frac{\mathbb{P}(A = 0 \mid M, W)\mathbb{P}(A = 1 \mid W)}{\mathbb{P}(A = 1 \mid M, W)\mathbb{P}(A = 0 \mid W)}.$$

Thus we can change the expression of the EIF a bit as follows. First, some more notation that will be useful later:

- Let $g(a \mid w)$ denote $\mathbb{P}(A = a \mid W = w)$
- Let $e(a \mid m, w)$ denote $\mathbb{P}(A = a \mid M = m, W = w)$
- Let $b(a, m, w)$ denote $\mathbb{E}(Y \mid A = a, M = m, W = w)$
- The EIF is

$$
\begin{aligned}
D(O) = & \left\{ \frac{I(A=1)}{g(0 \mid W)} \frac{e(0 \mid M, W)}{e(1 \mid M, W)} - \frac{I(A=0)}{g(0 \mid W)} \right\} \times [Y - b(A, M, W)] \\
& + \frac{I(A=0)}{g(0 \mid W)} \big\{ Q(M, W) - \mathbb{E}[Q(M, W) | W, A = 0] \big\} \\
& + \mathbb{E}[Q(M, W) | W, A = 0] - \psi(\mathbb{P})
\end{aligned}
$$

6.1.2 How to compute the one-step estimator (akin to Augmented IPW)

First we will generate some data:

```
mean_y <- function(m, a, w) abs(w) + a * m
mean_m <- function(a, w) plogis(w^2 - a)
pscore <- function(w) plogis(1 - abs(w))

w_big <- runif(1e6, -1, 1)
trueval <- mean((mean_y(1, 1, w_big) - mean_y(1, 0, w_big)) * mean_m(0, w_b
                 + (mean_y(0, 1, w_big) - mean_y(0, 0, w_big)) *
                   (1 - mean_m(0, w_big)))

n <- 1000
w <- runif(n, -1, 1)
a <- rbinom(n, 1, pscore(w))
m <- rbinom(n, 1, mean_m(a, w))
y <- rnorm(n, mean_y(m, a, w))
```

Recall that the one-step estimator is defined as the bias-corrected g-computation estimator:

$$\psi(\hat{\mathbb{P}}) + \frac{1}{n} \sum_{i=1}^{n} D(O; \hat{\mathbb{P}}_i)$$

Can be computed in the following steps:

1. Fit models for $g(a \mid w)$, $e(a \mid m, w)$, and $b(a, m, w)$
   - In this example we will use Generalized Additive Models [CITE] for tractability
   - In applied settings we recommend using an ensemble of data-adaptive regression algorithms, such as the Super Learner [CITE]

```r
library(mgcv)
## fit model for E(Y | A, W)
b_fit <- gam(y ~ m:a + s(w, by = a)
## fit model for P(A = 1 | M, W)
e_fit <- gam(a ~ m + w + s(w, by = m), family = binomial)
## fit model for P(A = 1 | W)
g_fit <- gam(a ~ w, family = binomial)
```

2. Compute predictions $g(1 \mid w)$, $g(0 \mid w)$, $e(1 \mid m, w)$, $e(0 \mid m, w)$,$b(1, m, w)$, $b(0, m, w)$, and $b(a, m, w)$

```r
## Compute P(A = 1 | W)
g1_pred <- predict(g_fit, type = 'response')
## Compute P(A = 0 | W)
g0_pred <- 1 - g1_pred
## Compute P(A = 1 | M, W)
e1_pred <- predict(e_fit, type = 'response')
## Compute P(A = 0 | M, W)
e0_pred <- 1 - e1_pred
## Compute E(Y | A = 1, M, W)
b1_pred <- predict(b_fit, newdata = data.frame(a = 1, m, w))
## Compute E(Y | A = 0, M, W)
b0_pred <- predict(b_fit, newdata = data.frame(a = 0, m, w))
## Compute E(Y | A, M, W)
b_pred  <- predict(b_fit)
```

3. Compute $Q(M, W)$, fit a model for $\mathbb{E}[Q(M, W)|W, A]$, and predict at $A = 0$

```
## Compute Q(M, W)
pseudo <- b1_pred - b0_pred
## Fit model for E[Q(M, W) | A, W]
q_fit <- gam(pseudo ~ a + w + s(w, by = a))
## Compute E[Q(M, W) | A = 0, W]
q_pred <- predict(q_fit, newdata = data.frame(a = 0, w = w))
```

4. Estimate the weights
$$\left\{\frac{I(A=1)}{g(0\mid W)}\frac{e(0\mid M,W)}{e(1\mid M,W)}-\frac{I(A=0)}{g(0\mid W)}\right\}$$
using the above predictions:

```
weights <- a / g0_pred * e0_pred / e1_pred - (1 - a) / g0_pred
```

5. Compute the uncentered EIF:

```
eif <- weights * (y - b_pred) + (1 - a) / g0_pred * (pseudo - q_pred) +
  q_pred
```

6. The one step estimator is the mean of the uncentered EIF

```
## One-step estimator
mean(eif)
```

6.1.3 Performance of the one-step estimator in a small simulation study

First, we create a wrapper around the estimator

```
one_step <- function(y, m, a, w) {
  b_fit <- gam(y ~ m:a + s(w, by = a))
  e_fit <- gam(a ~ m + w + s(w, by = m), family = binomial)
  g_fit <- gam(a ~ w, family = binomial)
  g1_pred <- predict(g_fit, type = 'response')
  g0_pred <- 1 - g1_pred
  e1_pred <- predict(e_fit, type = 'response')
  e0_pred <- 1 - e1_pred
  b1_pred <- predict(b_fit, newdata = data.frame(a = 1, m, w),
                  type = 'response')
  b0_pred <- predict(b_fit, newdata = data.frame(a = 0, m, w),
                      type = 'response')
  b_pred  <- predict(b_fit, type = 'response')
  pseudo <- b1_pred - b0_pred
  q_fit <- gam(pseudo ~ a + w + s(w, by = a))
  q_pred <- predict(q_fit, newdata = data.frame(a = 0, w = w))
  weights <- a / g0_pred * e0_pred / e1_pred - (1 - a) / g0_pred
  eif <- weights * (y - b_pred) + (1 - a) / g0_pred *
    (pseudo - q_pred) + q_pred
  return(mean(eif))
}
```

Let us first examine the bias

- The true value is:

```
w_big <- runif(1e6, -1, 1)
trueval <- mean((mean_y(1, 1, w_big) - mean_y(1, 0, w_big)) * mean_m(0
  (mean_y(0, 1, w_big) - mean_y(0, 0, w_big)) * (1 - mean_m(0, w_big))
print(trueval)
```

- Bias simulation

```
estimate <- numeric(1000)
for (i in 1:1000) {
  n <- 1000
  w <- runif(n, -1, 1)
  a <- rbinom(n, 1, pscore(w))
  m <- rbinom(n, 1, mean_m(a, w))
  y <- rnorm(n, mean_y(m, a, w))

  estimate[i] <- one_step(y, m, a, w)
}
```

```
hist(estimate)
abline(v = trueval, col = ”red”, lwd = 4)
```

- And now the confidence intervals:

```
cis <- cbind(
  estimate - qnorm(0.975) * sd(estimate),
  estimate + qnorm(0.975) * sd(estimate)
)

ord <- order(rowSums(cis))
lower <- cis[ord, 1]
upper <- cis[ord, 2]
curve(trueval + 0 * x,
  ylim = c(0, 1), xlim = c(0, 1001), lwd = 2, lty = 3, xaxt = ”n”,
  xlab = ””, ylab = ”Confidence␣interval”, cex.axis = 1.2, cex.lab = 1.2
)
for (i in 1:1000) {
  clr <- rgb(0.5, 0, 0.75, 0.5)
  if (upper[i] < trueval || lower[i] > trueval) clr <- rgb(1, 0, 0, 1)
  points(rep(i, 2), c(lower[i], upper[i]), type = ”l”, lty = 1, col = clr)
}
text(450, 0.10, ”n=1000␣repetitions␣=␣1000␣”, cex = 1.2)
text(450, 0.01, paste0(
  ”Coverage␣probability␣=␣”,
  mean(lower < trueval & trueval < upper), ”%”
), cex = 1.2)
```

Chapter 7

Estimating (in)direct effects with R packages

7.1 medshift: Stochastic (in)direct effects

We are interested in assessing the population intervention direct effect and the population intervention indirect effect, based on the effect decomposition of the population intervention effect introduced in Díaz and Hejazi (2020).

To proceed, we'll use as our running example a simple data set from an observational study of the relationship between BMI and kids behavior, distributed as part of the mma R package on CRAN[1]. First, let's load the packages we'll be using and set a seed; then, load this data set and take a quick look at it

```
# preliminaries
library(data.table)
library(dplyr)
library(sl3)
library(medshift)
library(mma)
set.seed(429153)

# load and examine data
data(weight_behavior)
```

[1] https://CRAN.R-project.org/package=mma

```
dim(weight_behavior)
head(weight_behavior)
```

The documentation for the data set describes it as a "database obtained from the Louisiana State University Health Sciences Center, New Orleans, by Dr. Richard Scribner. He explored the relationship between BMI and kids behavior through a survey at children, teachers and parents in Grenada in 2014. This data set includes 691 observations and 15 variables."

Unfortunately, the data set contains a few observations with missing values. As these are unrelated to the object of our analysis, we'll simply remove these for the time being. Note that in a real data analysis, we might consider strategies to fully make of the observed data, perhaps by imputing missing values. For now, we simply remove the incomplete observations, resulting in a data set with fewer observations but much the same structure as the original:

For the analysis of this observational data set, we focus on the effect of participating in a sports team (sports) on the BMI of children (bmi), taking several related covariates as mediators (snack, exercises, overweigh) and all other collected covariates as potential confounders. Considering an NPSEM, we separate the observed variables from the data set into their corresponding nodes as follows

```
Y <- weight_behavior_complete$bmi
A <- weight_behavior_complete$sports
Z <- weight_behavior_complete %>%
  select(snack, exercises, overweigh)
W <- weight_behavior_complete %>%
  select(
    age, sex, race, numpeople, car, gotosch, tvhours, cmpthours,
    cellhours, sweat
  )
```

Finally, in our analysis, we consider an incremental propensity score intervention (IPSI), as first proposed by **?**, wherein the *odds of participating in a sports team* is modulated by some fixed amount ($0 \leq \delta \leq \infty$) for each individual. Such an intervention may be interpreted as the effect of a school program that motivates children to participate in sports teams. To exemplify our approach, we postulate a motivational intervention that *triples the odds* of participating in a sports team for each individual:

```
delta_shift_ipsi <- 3
```

To easily incorporate ensemble machine learning into the estimation procedure, we rely on the facilities provided in the sl3 R package[2] (**?**). For a complete guide on using the sl3 R package, consider consulting https://tlverse.org/sl3, or https://tlverse.org (and https://github.com/tlverse) for the tlverse ecosystem, of which sl3 is a major part. We construct an ensemble learner using a handful of popular machine learning algorithms below

```
# SL learners used for continuous data (the nuisance parameter M)
xgb_contin_lrnr <- Lrnr_xgboost$new(nrounds = 50, objective = ”reg:linear”)
enet_contin_lrnr <- Lrnr_glmnet$new(
  alpha = 0.5, family = ”gaussian”,
  nfolds = 3
)
lasso_contin_lrnr <- Lrnr_glmnet$new(
  alpha = 1, family = ”gaussian”,
  nfolds = 3
)
fglm_contin_lrnr <- Lrnr_glm_fast$new(family = gaussian())
contin_lrnr_lib <- Stack$new(
  enet_contin_lrnr, lasso_contin_lrnr,
  fglm_contin_lrnr, xgb_contin_lrnr
)
sl_contin_lrnr <- Lrnr_sl$new(
  learners = contin_lrnr_lib,
  metalearner = Lrnr_nnls$new()
)

# SL learners used for binary data (nuisance parameters G and E in this cas
xgb_binary_lrnr <- Lrnr_xgboost$new(nrounds = 50, objective = ”reg:logistic
enet_binary_lrnr <- Lrnr_glmnet$new(
  alpha = 0.5, family = ”binomial”,
  nfolds = 3
)
lasso_binary_lrnr <- Lrnr_glmnet$new(
  alpha = 1, family = ”binomial”,
  nfolds = 3
)
```

[2] https://tlverse.org/sl3

```
fglm_binary_lrnr <- Lrnr_glm_fast$new(family = binomial())
binary_lrnr_lib <- Stack$new(
  enet_binary_lrnr, lasso_binary_lrnr,
  fglm_binary_lrnr, xgb_binary_lrnr
)
logistic_metalearner <- make_learner(
  Lrnr_solnp,
  metalearner_logistic_binomial,
  loss_loglik_binomial
)
sl_binary_lrnr <- Lrnr_sl$new(
  learners = binary_lrnr_lib,
  metalearner = logistic_metalearner
)
```

7.1.1 Decomposing the population intervention effect

We may decompose the population intervention effect (PIE) in terms of a *population intervention direct effect* (PIDE) and a *population intervention indirect effect* (PIIE):

$$\overbrace{\mathbb{E}\{Y(A_\delta, Z(A_\delta)) - Y(A_\delta, Z)\}}^{\text{PIIE}} + \overbrace{\mathbb{E}\{Y(A_\delta, Z) - Y(A, Z)\}}^{\text{PIDE}} .$$

This decomposition of the PIE as the sum of the population intervention direct and indirect effects has an interpretation analogous to the corresponding standard decomposition of the average treatment effect. In the sequel, we will compute each of the components of the direct and indirect effects above using appropriate estimators as follows

- For $\mathbb{E}\{Y(A, Z)\}$, the sample mean $\frac{1}{n}\sum_{i=1}^{n} Y_i$ is sufficient;
- for $\mathbb{E}\{Y(A_\delta, Z)\}$, an efficient one-step estimator for the effect of a joint intervention altering the exposure mechanism but not the mediation mechanism, as proposed in Díaz and Hejazi (2020); and,
- for $\mathbb{E}\{Y(A_\delta, Z_{A_\delta})\}$, an efficient one-step estimator for the effect of a joint intervention altering both the exposure and mediation mechanisms, as proposed in **?** and implemented in the npcausal R package[3].

[3] https://github.com/ehkennedy/npcausal

7.1.2 Estimating the effect decomposition term

As given in Díaz and Hejazi (2020), the statistical functional identifying the decomposition term that appears in both the PIDE and PIIE $\mathbb{E}\{Y(A_\delta, Z)\}$, which corresponds to altering the exposure mechanism while keeping the mediation mechanism fixed, is

$$\theta_0(\delta) = \int m_0(a, z, w) g_{0,\delta}(a \mid w) p_0(z, w) d\nu(a, z, w),$$

for which a one-step estimator is available. The corresponding *efficient influence function* (EIF) with respect to the nonparametric model $\mathcal{M}$ is $D_{\eta,\delta}(o) = D^Y_{\eta,\delta}(o) + D^A_{\eta,\delta}(o) + D^{Z,W}_{\eta,\delta}(o) - \theta(\delta)$. The one-step estimator may be computed using the EIF estimating equation, making use of cross-fitting (**??**) to circumvent any need for entropy conditions (i.e., Donsker class restrictions). The resultant estimator is

$$\hat{\theta}(\delta) = \frac{1}{n} \sum_{i=1}^{n} D_{\hat{\eta}_{j(i)},\delta}(O_i) = \frac{1}{n} \sum_{i=1}^{n} \left\{ D^Y_{\hat{\eta}_{j(i)},\delta}(O_i) + D^A_{\hat{\eta}_{j(i)},\delta}(O_i) + D^{Z,W}_{\hat{\eta}_{j(i)},\delta}(O_i) \right\},$$

which is implemented in the medshift R package. We make use of that implementation to estimate $\mathbb{E}\{Y(A_\delta, Z)\}$ via its one-step estimator $\hat{\theta}(\delta)$ below

```
# let’s compute the parameter where A (but not Z) are shifted
theta_eff <- medshift(
  W = W, A = A, Z = Z, Y = Y,
  delta = delta_shift_ipsi,
  g_learners = sl_binary_lrnr,
  e_learners = sl_binary_lrnr,
  m_learners = sl_contin_lrnr,
  phi_learners = Lrnr_hal9001$new(),
  estimator = ”onestep”,
  estimator_args = list(cv_folds = 3)
)
summary(theta_eff)
```

7.1.3 Estimating the direct effect

Recall that, based on the decomposition outlined previously, the population intervention direct effect may be denoted $\beta_{\text{PIDE}}(\delta) = \theta_0(\delta) - \mathbb{E}Y$. Thus, an estimator of the PIDE, $\hat{\beta}_{\text{PIDE}}(\delta)$ may be expressed as a composition of estimators of its constituent parameters:

$$\hat{\beta}_{\text{PIDE}}(\delta) = \hat{\theta}(\delta) - \frac{1}{n} \sum_{i=1}^{n} Y_i.$$

Based on the above, we may construct an estimator of the PIDE using quantities already computed. The convenience function below applies the simple delta method required in the case of a linear contrast between the two constituent parameters:

```
# convenience function to compute inference via delta method: EY1 - EY
linear_contrast <- function(params, eifs, ci_level = 0.95) {
  # bounds for confidence interval
  ci_norm_bounds <- c(-1, 1) * abs(stats::qnorm(p = (1 - ci_level) / 2
  param_est <- params[[1]] - params[[2]]
  eif <- eifs[[1]] - eifs[[2]]
  se_eif <- sqrt(var(eif) / length(eif))
  param_ci <- param_est + ci_norm_bounds * se_eif
  # parameter and inference
  out <- c(param_ci[1], param_est, param_ci[2])
  names(out) <- c(”lwr_ci”, ”param_est”, ”upr_ci”)
  return(out)
}
```

With the above convenience function in hand, we'll construct or extract the necessary components from existing objects and simply apply the function:

```
# parameter estimates and EIFs for components of direct effect
EY <- mean(Y)
eif_EY <- Y - EY
params_de <- list(theta_eff$theta, EY)
eifs_de <- list(theta_eff$eif, eif_EY)

# direct effect = EY - estimated quantity
de_est <- linear_contrast(params_de, eifs_de)
de_est
```

7.2 `medoutcon`: Natural/Interventional Effects

An exposure of interest often affects an outcome directly, or indirectly by the mediation of some intermediate variables. Identifying and quantifying the mechanisms underlying causal effects is an increasingly popular endeavor in public health, medicine, and the social sciences, as knowledge of such mechanisms can improve understanding of both *why and how* treatments can be effective. Such mechanistic knowledge may be arguably even more important in cases where treatments result in unanticipated ineffective or even harmful effects.

Traditional techniques for mediation analysis fare poorly in the face of intermediate confounding. Classical parameters like the natural (in)direct effects face a lack of identifiability in cases where mediator-outcome (i.e., intermediate) confounders affected by exposure complicate the relationship between the exposure, mediators, and outcome. **?** provide a theoretical and computational study of the properties of newly developed interventional (in)direct effect estimands within the non-parametric statistical model. Among their contributions, **?**

- derive the efficient influence function (EIF), an key object in semiparametric efficiency theory;
- use the EIF to develop two asymptotically optimal, non-parametric estimators, each of which is capable of leveraging machine learning for the estimation of nuisance parameters; and
- present theoretical conditions under which their proposed estimators are consistent, multiply robust, and efficient.

7.2.1 Problem Setup and Notation

The problem addressed by the work of **?** may be represented by the following nonparametric structural equation model (NPSEM):

$$W = f_W(U_W); A = f_A(W, U_A); Z = f_Z(W, A, U_Z);$$
$$M = f_M(W, A, Z, U_M); Y = f_Y(W, A, Z, M, U_Y).$$

In the NPSEM, W denotes a vector of observed pre-treatment covariates, A denotes a categorical treatment variable, Z denotes an intermediate confounder affected by treatment, M denotes a (possibly multivariate) mediator, and Y denotes a continuous or binary outcome. The vector of exogenous factors $U = (U_W, U_A, U_Z, U_M, U_Y)$, and the functions f, are assumed deterministic but unknown. Importantly, the NPSEM encodes a time-ordering between these variables and allows the evaluation of counterfactual quantities defined by intervening on a set of nodes of the NPSEM. The observed data unit can be represented by the random variable $O = (W, A, Z, M, Y)$; we consider access to $O_1, \ldots, O_n$, a sample of n i.i.d. observations of O.

? additionally define the following parameterizations, familiarity with which will be useful for using the medoutcon R package[4]. In particular, these authors define $g(a \mid w)$ as

[4] https://github.com/nhejazi/medoutcon

the probability mass function of $A = a$ conditional on $W = w$ and use $h(a \mid m, w)$ to denote the probability mass function of $A = a$ conditional on $(M, W) = (m, w)$. Further, **?** use $b(a, z, m, w)$ to denote the outcome regression function $\mathbb{E}(Y \mid A = a, Z = z, M = m, W = w)$, as well as $q(z \mid a, w)$ and $r(z \mid a, m, w)$ to denote the corresponding conditional densities of Z.

7.2.2 Interventional (In)Direct Effects

? define the *total effect* of A on Y in terms of a contrast between two user-supplied values $a', a^\star \in \mathcal{A}$. Examination of the NPSEM reveals that there are four paths involved in this effect, namely $A \rightarrow Y$, $A \rightarrow M \rightarrow Y$, $A \rightarrow Z \rightarrow Y$, and $A \rightarrow Z \rightarrow M \rightarrow Y$. Mediation analysis has classically considered the *natural direct effect* (NDE) and the *natural indirect effect* (NIE), which are defined as $\mathbb{E}_c(Y_{a',M_{a^\star}} - Y_{a^\star,M_{a^\star}})$ and $\mathbb{E}_c(Y_{a',M_{a'}} - Y_{a',M_{a^\star}})$, respectively. The natural direct effect measures the effect through paths *not* involving the mediator ($A \rightarrow Y$ and $A \rightarrow Z \rightarrow Y$), whereas the natural indirect effect measures the effect through paths involving the mediator ($A \rightarrow M \rightarrow Y$ and $A \rightarrow Z \rightarrow M \rightarrow Y$). As the sum of the natural direct and indirect effects equals the average treatment effect $\mathbb{E}_c(Y_1 - Y_0)$, this effect decomposition is appealing. Unfortunately, the natural direct and indirect effects are not generally identified in the presence of an intermediate confounder affected by treatment.

To circumvent this issue, **?** define the direct and indirect effects using stochastic interventions on the mediator, following a strategy previously outlined by **?** and **?**, among others. Let G_a denote a random draw from the conditional distribution of M_a conditional on W. Consider the effect of A on Y defined as the difference in expected outcome in hypothetical worlds in which $(A, M) = (a', G_{a'})$ versus $(A, M) = (a^\star, G_{a^\star})$ with probability one, which may be decomposed into direct and indirect effects as follows

$$\mathbb{E}_c(Y_{a',G_{a'}} - Y_{a^\star,G_{a^\star}}) = \underbrace{\mathbb{E}_c(Y_{a',G_{a'}} - Y_{a',G_{a^\star}})}_{\text{Indirect effect (through } M)} + \underbrace{\mathbb{E}_c(Y_{a',G_{a^\star}} - Y_{a^\star,G_{a^\star}})}_{\text{Direct effect (not through } M)}.$$

Like the natural direct effect, this interventional direct effect measures the effects through paths not involving the mediator. Likewise, the interventional indirect effect measures the effect through paths involving the mediator. Note, however, that natural and interventional mediation effects have different interpretations. That is, the interventional indirect effect measures the effect of fixing the exposure at a' while setting the mediator to a random draw $G_{a^\star}$ from those with exposure a' versus a random draw $G_{a'}$ from those with exposure $a^\star$, given covariates W. As is clear from the effect decomposition, the term $\theta_c = \mathbb{E}_c(Y_{a',G_{a^\star}})$ is required for estimation of both the interventional direct and indirect effects; thus, **?** focus on estimation of this quantity. Importantly, it has been shown that θ_c is identified by the statistical functional

$$\theta = \int b(a', z, m, w) q(z \mid a', w) p(m \mid a^{\star}, w) p(w) d\nu(w, z, m)$$

under a set of standard identifiability conditions (**?**), which are further reviewed in **?**.

7.2.3 Efficient Estimation

? define two efficient estimators of their interventional (in)direct effects. These are based on the one-step estimation and targeted minimum loss (TML) estimation frameworks, respectively. Briefly, both estimation strategies proceed in two stages, starting by first constructing initial estimates of the nuisance parameters present in the EIF, then proceeding to apply distinct bias-correction strategies in their second stages. Both estimation strategies require an assumption about the behavior of initial estimators of the nuisance parameters (specifically, that these lie in a Donsker class); however, the need for such an assumption may be avoided by making use of cross-validation in the fitting fo initial estimators. The medoutcon R package requires the use of cross-validation in the construction of these initial estimates, resulting in cross-fitted one-step and and cross-validated TML estimators (**???**).

The one-step estimator $\hat{\theta}_{\text{os}}$ is constructed by adding the empirical mean of the EIF (evaluated at initial estimates of the nuisance parameters) to the substitution estimator. By constrast, the TML estimator $\hat{\theta}_{\text{tmle}}$ updates the components of the substitution estimator via logistic tilting models formulated to ensure that relevant score equations appearing in the EIF are (approximately) solved. While the estimators are asymptotically equivalent, TML estimators have been shown to exhibit superior finite-sample performance, making them potentially more reliable than one-step estimators. For the exact form of the EIF as well as those of the one-step and TML estimators, consult **?**.

7.2.4 Data Analysis Example

7.2.4.1 Setting up the data example

Now, we'll take a look at how to estimate the interventional direct and indirect effects using a simulated data example. **?** illustrate the use of their estimators of these effects in an application in which they seek to elucidate the mechanisms behind the unintended harmful effects that a housing intervention had on adolescent girls' risk behavior.

First, let's load a few required packages and set a seed for our simulation.

```
library(data.table)
library(medoutcon)
library(sl3)
set.seed(75681)
n_obs <- 500
```

Next, we'll generate a very simple simulated dataset. The function make_example_data, defined below, generates three binary baseline covariates $W = (W_1, W_2, W_3)$, a binary exposure variable A, a single binary mediateor M an intermediate confounder Z that affects the mediator M and is itself affected by the exposure A, and, finally, a binary outcome Y that is a function of (W, A, Z, M).

```
# produces a simple data set based on ca causal model with mediation
make_example_data <- function(n_obs = 1000) {
  ## baseline covariates
  w_1 <- rbinom(n_obs, 1, prob = 0.6)
  w_2 <- rbinom(n_obs, 1, prob = 0.3)
  w_3 <- rbinom(n_obs, 1, prob = pmin(0.2 + (w_1 + w_2) / 3, 1))
  w <- cbind(w_1, w_2, w_3)
  w_names <- paste(”W”, seq_len(ncol(w)), sep = ”_”)

  ## exposure
  a <- as.numeric(rbinom(n_obs, 1, plogis(rowSums(w) - 2)))

  ## mediator-outcome confounder affected by treatment
  z <- rbinom(n_obs, 1, plogis(rowMeans(-log(2) + w - a) + 0.2))

  ## mediator -- could be multivariate
  m <- rbinom(n_obs, 1, plogis(rowSums(log(3) * w[, -3] + a - z)))
  m_names <- ”M”

  ## outcome
  y <- rbinom(n_obs, 1, plogis(1 / (rowSums(w) - z + a + m)))

  ## construct output
  dat <- as.data.table(cbind(w = w, a = a, z = z, m = m, y = y))
  setnames(dat, c(w_names, ”A”, ”Z”, m_names, ”Y”))
  return(dat)
}
```

```
# set seed and simulate example data
example_data <- make_example_data(n_obs)
w_names <- stringr::str_subset(colnames(example_data), ”W’)
m_names <- stringr::str_subset(colnames(example_data), ”M’)
```

Now, let’s take a quick look at our simulated data:

```
# quick look at the data
head(example_data)
```

As noted above, all covariates in our dataset are binary; however, note that this need not be the case for using our methodology — in particular, the only current limitation is that the intermediate confounder Z must be binary when using our implemented TML estimator of the (in)direct effects.

Using this dataset, we’ll proceed to estimate the interventional (in)direct effects. In order to do so, we’ll need to estimate several nuisance parameters, including the exposure mechanism $g(A \mid W)$, a re-parameterized exposure mechanism that conditions on the mediators $h(A \mid M, W)$, the outcome mechanism $b(Y \mid M, Z, A, W)$, and two variants of the intermediate confounding mechanism $q(Z \mid A, W)$ and $r(Z \mid M, A, W)$. In order to estimate each of these nuisance parameters flexibly, we’ll rely on data adaptive regression strategies in order to avoid the potential for (parametric) model misspecification.

7.2.4.2 Ensemble learning of nuisance functions

As we’d like to rely on flexible, data adaptive regression strategies for estimating each of the nuisance parameters (g, h, b, q, r), we require a method for choosing among or combining the wide variety of available regression strategies. For this, we recommend the use of the Super Learner algorithm for ensemble machine learning (van der Laan et al., 2007). The recently developed sl3 R package[5] (**?**) provides a unified interface for deploying a wide variety of machine learning algorithms (simply called *learners* in the sl3 nomenclature) as well as for constructing Super Learner ensemble models of such learners. For a complete guide on using the sl3 R package, consider consulting `https://tlverse.org/sl3`, or `https://tlverse.org` (and `https://github.com/tlverse`) for the tlverse ecosystem, of which sl3 is an integral part.

To construct an ensemble learner using a handful of popular machine learning algorithms, we’ll first instantiate variants of learners from the appropriate classes for each algorithm,

[5] `https://tlverse.org/sl3`

and then create a Super Learner ensemble via the Lrnr_sl class. Below, we demonstrate the construction of an ensemble learner based on a modeling library including an intercept model, a main-terms GLM, ℓ_1-penalized Lasso regression, an elastic net regression that equally weights the ℓ_1 and ℓ_2 penalties, random forests (ranger), and the highly adaptive lasso (HAL):

```
# instantiate learners
mean_lrnr <- Lrnr_mean$new()
fglm_lrnr <- Lrnr_glm_fast$new(family = binomial())
lasso_lrnr <- Lrnr_glmnet$new(alpha = 1, family = ”binomial”, nfolds =
enet_lrnr <- Lrnr_glmnet$new(alpha = 0.5, family = ”binomial”, nfolds
rf_lrnr <- Lrnr_ranger$new(num.trees = 200)

# for HAL, use linear probability formulation, with bounding in unit i
hal_gaussian_lrnr <- Lrnr_hal9001$new(
  family = ”gaussian”,
  fit_control = list(
    max_degree = 3,
    n_folds = 3,
    use_min = TRUE,
    type.measure = ”mse”
  )
)
bound_lrnr <- Lrnr_bound$new(bound = 1e-6)
hal_bounded_lrnr <- Pipeline$new(hal_gaussian_lrnr, bound_lrnr)

# create learner library and instantiate super learner ensemble
lrnr_lib <- Stack$new(
  mean_lrnr, fglm_lrnr, enet_lrnr, lasso_lrnr,
  rf_lrnr, hal_bounded_lrnr
)
sl_lrnr <- Lrnr_sl$new(learners = lrnr_lib, metalearner = Lrnr_nnls$ne
```

While we recommend the use of a Super Learner ensemble model like the one constructed above in practice, such a library will be too computationally intensive for our examples. To reduce computation time, we construct a simpler library, using only a subset of the above learning algorithms:

```
# create simpler learner library and instantiate super learner ensembl
lrnr_lib <- Stack$new(mean_lrnr, fglm_lrnr, lasso_lrnr, rf_lrnr)
```

```
sl_lrnr <- Lrnr_sl$new(learners = lrnr_lib, metalearner = Lrnr_nnls$new())
```

Having set up our ensemble learner, we're now ready to estimate each of the interventional effects using the efficient estimators exposed in the medoutcon package.

7.2.4.3 Estimating the direct effect

We're now ready to estimate the interventional direct effect. This direct effect is computed as a contrast between the interventions $(a' = 1, a^\star = 0)$ and $(a' = 0, a^\star = 0)$. In particular, our efficient estimators of the interventional direct effect proceed by constructing estimators $\hat{\theta}(a' = 1, a^\star = 0)$ and $\hat{\theta}(a' = 0, a^\star = 0)$. Then, an efficient estimator of the direct effect is available by application of the delta method, that is, $\hat{\theta}^{\text{DE}} = \hat{\theta}(a' = 1, a^\star = 0) - \hat{\theta}(a' = 0, a^\star = 0)$. Applying the same principle to the EIF estimates, one can derive variance estimates and construct asymptotically correct Wald-style confidence intervals for $\hat{\theta}^{\text{DE}}$.

The medoutcon package makes the estimation task quite simple, as only a single call to the eponymous medoutcon function is required. As demonstrated below, we need only feed in each component of the observed data $O = (W, A, Z, M, Y)$ (of which W and M can be multivariate), specify the effect type, and the estimator. Additionally, for each nuisance parameter we may specify a separate regression function — in the examples below, we use the simpler Super Learner ensemble constructed above for fitting each nuisance function, but this need not be the case (i.e., different estimators may be used for each nuisance function).

First, we examine the one-step estimator of the interventional direct effect. Recall that the one-step estimator is constructed by adding the mean of the EIF (evaluated at initial estimates of the nuisance parameters) to the substitution estimator. As noted above, this is done separately for each of the two contrasts $(a' = 0, a^\star = 0)$ and $(a' = 1, a^\star = 0)$. Thus, the one-step estimator of this direct effect is constructed by application of the delta method to each of the one-step estimators (and EIFs) for these contrasts.

```
# compute one-step estimate of the interventional direct effect
os_de <- medoutcon(
  W = example_data[, ..w_names],
  A = example_data$A,
  Z = example_data$Z,
  M = example_data[, ..m_names],
  Y = example_data$Y,
  g_learners = sl_lrnr,
  h_learners = sl_lrnr,
  b_learners = sl_lrnr,
  q_learners = sl_lrnr,
  r_learners = sl_lrnr,
  effect = "direct",
  estimator = "onestep",
  estimator_args = list(cv_folds = 2)
)
summary(os_de)
```

Next, let's compare the one-step estimate to the TML estimate. Analogous to the case of the one-step estimator, the TML estimator can be evaluated via a single call to the medoutcon function:

```
# compute targeted minimum loss estimate of the interventional direct
tmle_de <- medoutcon(
  W = example_data[, ..w_names],
  A = example_data$A,
  Z = example_data$Z,
  M = example_data[, ..m_names],
  Y = example_data$Y,
  g_learners = sl_lrnr,
  h_learners = sl_lrnr,
  b_learners = sl_lrnr,
  q_learners = sl_lrnr,
  r_learners = sl_lrnr,
  effect = "direct",
  estimator = "tmle",
  estimator_args = list(cv_folds = 2, max_iter = 5)
)
summary(tmle_de)
```

Here, we recall that the TML estimator generally exhibits better finite-sample performance than the one-step estimator (van der Laan and Rose, 2011, 2018), so the TML estimate is likely to be more reliable in modest (realistic) sample sizes.

7.2.4.4 Estimating the indirect effect

Estimation of the interventional indirect effect proceeds similarly to the strategy discussed above for the corresponding direct effect. An efficient estimator can be computed as a contrast between the interventions $(a' = 1, a^\star = 0)$ and $(a' = 1, a^\star = 1)$. Specifically, our efficient estimators of the interventional indirect effect proceed by constructing estimators $\hat{\theta}(a' = 1, a^\star = 0)$ and $\hat{\theta}(a' = 1, a^\star = 1)$. Then, application of the delta method yields an efficient estimator of the indirect effect, that is, $\hat{\theta}^{\text{IE}} = \hat{\theta}(a' = 1, a^\star = 0) - \hat{\theta}(a' = 1, a^\star = 1)$. The same principle may be applied to the EIF estimates to derive variance estimates and construct asymptotically correct Wald-style confidence intervals for $\hat{\theta}^{\text{IE}}$.

Now, we examine the one-step estimator of the interventional indirect effect. The one-step estimator is constructed by adding the mean of the EIF (evaluated at initial estimates of the nuisance parameters) to the substitution estimator. As noted above, this is done separately for each of the two contrasts $(a' = 1, a^\star = 1)$ and $(a' = 1, a^\star = 0)$. Thus, the one-step estimator of this indirect effect is constructed by application of the delta method to each of the one-step estimators (and EIFs) for the contrasts.

```
# compute one-step estimate of the interventional indirect effect
os_ie <- medoutcon(
  W = example_data[, ..w_names],
  A = example_data$A,
  Z = example_data$Z,
  M = example_data[, ..m_names],
  Y = example_data$Y,
  g_learners = sl_lrnr,
  h_learners = sl_lrnr,
  b_learners = sl_lrnr,
  q_learners = sl_lrnr,
  r_learners = sl_lrnr,
  effect = ”indirect”,
  estimator = ”onestep”
)
summary(os_ie)
```

As before, let’s compare the one-step estimate to the TML estimate. Analogous to the case of the one-step estimator, the TML estimator can be evaluated via a single call to the medoutcon function, as demonstrated below

```
# compute targeted minimum loss estimate of the interventional indirect eff
```

```
tmle_ie <- medoutcon(
  W = example_data[, ..w_names],
  A = example_data$A,
  Z = example_data$Z,
  M = example_data[, ..m_names],
  Y = example_data$Y,
  g_learners = sl_lrnr,
  h_learners = sl_lrnr,
  b_learners = sl_lrnr,
  q_learners = sl_lrnr,
  r_learners = sl_lrnr,
  effect = ”indirect”,
  estimator = ”tmle”
)
summary(tmle_ie)
```

As before, the TML estimator provides better finite-sample performance than the one-step estimator, so it may be preferred in this example.

Bibliography

Díaz, I. and Hejazi, N. S. (2020). Causal mediation analysis for stochastic interventions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(3):661–683.

Hejazi, N. S., Rudolph, K. E., van der Laan, M. J., and Díaz, I. (2020). Nonparametric causal mediation analysis for stochastic interventional (in) direct effects. *arXiv preprint arXiv:2009.06203*.

Kennedy, E. H. (2018). Nonparametric causal effects based on incremental propensity score interventions. *Journal of the American Statistical Association*, (just-accepted).

Rudolph, K., Diaz, I., Hejazi, N., van der Laan, M., Luo, S., Shulman, M., Campbell, A., Rotrosen, J., and Nunes, E. (2020). Explaining differential effects of medication for opioid use disorder using a novel approach incorporating mediating variables. *Addiction*.

van der Laan, M. J., Polley, E. C., and Hubbard, A. E. (2007). Super Learner. *Statistical Applications in Genetics and Molecular Biology*, 6(1).

van der Laan, M. J. and Rose, S. (2011). *Targeted learning: causal inference for observational and experimental data*. Springer Science & Business Media.

van der Laan, M. J. and Rose, S. (2018). *Targeted Learning in Data Science: Causal Inference for Complex Longitudinal Studies*. Springer Science & Business Media.