

Statistical Learning in Mediation Analysis

Lab for Chapter 3: Estimating natural direct and indirect effects in R

David Benkeser
Emory University

Iván Díaz
New York University

Marco Carone
University of Washington

MODULE 14

**Summer Institute in Statistics for
Clinical and Epidemiological Research**

July 2024

Contents of this lab

- 1 Illustration of estimation of the NDE and NIE using the [medoutcon R package](#)
 - [medoutcon](#) seamlessly integrates Super Learning via the implementation in the [s13 R package](#)
 - The material in this lab is based on [this](#) online resource
- 2 Do-it-yourself analysis of a real dataset

Illustrative dataset

We will use the dataset `weight_behavior` from the [mma R package](#). 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.

Setting up the dataset

The data set contains some observations with missing values. Here we remove those observations to simplify the demonstration of the mediation methods. In practice we recommend using appropriate corrections (e.g., imputation, inverse weighting) to fully take advantage of the observed data.

```
library(mma)
library(tidyverse)
# load and examine data
data(weight_behavior)
dim(weight_behavior)

## [1] 691 15

# drop missing values
weight_behavior <- weight_behavior %>%
  drop_na() %>%
  as_tibble()
weight_behavior$sports <- as.numeric(weight_behavior$sports) - 1
```

Setting up the problem

We focus on the causal effects of participating in a sports team (`sports`) on the BMI of children (`bmi`), taking into consideration several mediators (`snack`, `exercises`, `overweigh`). All other measured covariates are taken to be potential baseline confounders.

```
summary(weight_behavior %>% select(sports, bmi, snack, exercises, overweigh))
```

```
##      sports          bmi      snack      exercises
## Min.   :0.0000   Min.   :11.85   1:501   Min.    : 0.000
## 1st Qu.:0.0000   1st Qu.:16.65   2: 66   1st Qu.:  4.000
## Median :0.0000   Median :18.09           Median :  8.000
## Mean   :0.4092   Mean   :19.13           Mean   :  9.392
## 3rd Qu.:1.0000   3rd Qu.:20.51           3rd Qu.:12.000
## Max.   :1.0000   Max.   :38.03           Max.   :180.000
##      overweigh
## Min.   :0.00000
## 1st Qu.:0.00000
## Median :0.00000
## Mean   :0.08642
## 3rd Qu.:0.00000
## Max.   :1.00000
```

Why use Super Learning?

- As discussed in the workshop, estimating the relevant regression functions using flexible regression reduces the chances of model misspecification and leads to more correct estimates
- Knowing a-priori which flexible regression algorithm is best can be challenging. Super Learning aids in this task by building a convex combination of user-supplied algorithms... this convex combination has been proved to be optimal asymptotically (van der Laan, Polley, and Hubbard 2007)
- `medoutcon` uses the Super Learner implementation in the `s13` package.

Setting up s13

Let us start by loading the `s13` library and initializing an ensemble learner based on a main terms generalized linear model (GLM), an ℓ_1 -penalized GLM, and random forests:

```
library(s13)
# instantiate learners
fglm_lrnr <- Lrnr_glm_fast$new()
lasso_lrnr <- Lrnr_glmnet$new(alpha = 1, nfolds = 3)
rf_lrnr <- Lrnr_ranger$new(num.trees = 200)
# create learner library and instantiate super learner ensemble
lrnr_lib <- Stack$new(fglm_lrnr, lasso_lrnr, rf_lrnr)
s1_lrnr <- Lrnr_sl$new(learners = lrnr_lib, metalearner = Lrnr_nnls$new())
```

Cross-fitting

- In addition to performing Super Learning, the estimators implemented in `medoutcon` are *cross-fitted*
- Cross-fitting is similar to cross-validation in that it is a sample splitting procedure
- Unlike cross-validation, the goal of cross-fitting is simply to obtain out-of-sample predictions, not to validate the algorithms
- Cross-fitting is known to improve the performance of the estimators and allow the use of more flexible regression techniques

Introducing the main function `medoutcon()`

The package may be installed running

```
library(devtools)
install_github('nhejazi/medoutcon')
```

The workhorse of the package is the function `medoutcon()`. This function takes at minimum the following arguments:

- `W`: a data frame with baseline confounders
- `A`: a binary (zero or one) treatment variable
- `Z`: an intermediate confounder (we will explain this in Chapter 4, please ignore it for now and set it to `NULL`)
- `Y`: binary or continuous outcome vector
- `g_learners`: an `s13` learner stack for $P(A = a \mid W = w)$
- `h_learners`: an `s13` learner stack for $P(A = a \mid M = m, W = w)$
- `b_learners`: an `s13` learner stack for $E(Y \mid A = 1, M = m, W = w)$
- `effect`: which effect is to be estimated “direct” or “indirect”
- `estimator`: which estimator is to be used “onestep” or “tmle”
- `estimator_args`: other estimation parameters such as the number of cross-fitting folds

Estimating the natural direct effect

```
library(medoutcon)
# compute one-step estimate of the natural direct effect
nde_onestep <- medoutcon(
  W = weight_behavior[, c("age", "sex", "race", "tvhours")],
  A = weight_behavior$sports,
  Z = NULL,
  M = weight_behavior[, c("snack", "exercises", "overweigh")],
  Y = weight_behavior$bmi,
  g_learners = lasso_lrnr,
  h_learners = lasso_lrnr,
  b_learners = lasso_lrnr,
  effect = "direct",
  estimator = "onestep",
  estimator_args = list(cv_folds = 5)
)
```

Estimating the natural indirect effect

```
# compute one-step estimate of the natural indirect effect
nie_onestep <- medoutcon(
  W = weight_behavior[, c("age", "sex", "race", "tvhours")],
  A = (as.numeric(weight_behavior$sports) - 1),
  Z = NULL,
  M = weight_behavior[, c("snack", "exercises", "overweigh")],
  Y = weight_behavior$bmi,
  g_learners = lasso_lrnr,
  h_learners = lasso_lrnr,
  b_learners = lasso_lrnr,
  effect = "indirect",
  estimator = "onestep",
  estimator_args = list(cv_folds = 5)
)
```

Results

```
summary(nde_onestep)
## # A tibble: 1 x 7
##   lwr_ci param_est upr_ci var_est eif_mean estimator param
##   <dbl>   <dbl> <dbl>  <dbl>   <dbl> <chr>    <chr>
## 1 -0.465   -0.0117  0.442  0.0535     0 onestep direct~

summary(nie_onestep)
## # A tibble: 1 x 7
##   lwr_ci param_est upr_ci var_est eif_mean estimator param
##   <dbl>   <dbl> <dbl>  <dbl>   <dbl> <chr>    <chr>
## 1  0.448    0.999  1.55  0.0792 5.88e-16 onestep indire~
```

We can therefore conclude that the effect of participation on a sports team on BMI operates primarily through the variables `snack`, `exercises`, and `overweigh`.

Do-it-yourself analysis of a real dataset

- 1 Consider the dataset framing available in the `mediation` R package
- 2 This dataset is from a nationally representative randomized experiment on how the framing of media discourse shapes public opinion and immigration policy (Brader et al., 2008)
- 3 The authors conducted a randomized experiment in which 265 subjects are exposed to different media stories about immigration
- 4 Please read the description of the data in the R package and the original research article, and analyze the data to answer the following question:

To what extent is the causal effect of the tone of the story on negative attitude towards immigration mediated by anxiety?

- 5 Feel free to make decisions about confounders, mediators, etc. based on the above information and your knowledge of the problem.

References and additional reading

References:

Ted Brader, Nicholas A Valentino, and Elizabeth Suhay. What triggers public opposition to immigration? Anxiety, group cues, and immigration threat. *American Journal of Political Science*, 52(4): 959–978, 2008.