Controlled Direct Effects for Unmasking

Gabriel Loewinger, Mats Stensrud, Alex Levis

2025-09-17

lmtp R package

We use the `lmtp` R package. In addition to the two papers in the references section below, the following resources may also be helpful:

- https://muse.jhu.edu/pub/56/article/883479/pdf (https://muse.jhu.edu/pub/56/article/883479/pdf)
- https://beyondtheate.com/ (https://beyondtheate.com/)

Data Formating

Let's load the data, look at the format, and work through a simple implementation. We simulated this data for demonstration purposes. The code we used to simulate the data is available on our Github repo, although the details are less important than understanding the steps in the analysis code below so one can apply a similar analysis on a real dataset.

```
dat <- read.csv("CDE_data.csv")
head(dat)
```

```
      A     E     B        SS    mf    age     psych_h           Y    id
  <int> <int> <int>     <dbl> <int>  <dbl>       <dbl>       <dbl> <int>
1     0     1     0  0.000000     0   0.50  0.03670436  -1.4141453     1
2     1     1     1  1.095147     0   0.70  0.20374381   2.1202515     2
3     0     1     1  0.000000     1   0.90  0.51378114  -0.7495239     3
4     1     1     1  3.070751     0   0.68  0.23633989   1.5525878     4
5     1     1     1  1.150120     1   0.76  0.57543567   1.7298167     5
6     0     1     0  0.000000     1   0.92  0.48225791  -1.7042723     6
6 rows
```

```
dat <- as.data.frame(dat) # set to data.frame
```

Each row is a participant. The columns contain the following variables:

- `A` is a binary indicator of treatment/control
- `E` is the post-treatment Expectancy measurement
- `B` is the post-treatment Belief measurement
- `SS` is a set/setting variable measured during treatment administration
- `mf` is male/female biological sex
- `age` is participant age

- `psych_h` is psychedelic history/experience prior to the study
- `Y` is the outcome (e.g., depression levels)
- `id` is the subject ID

Controlled Direct Effect

We use the counterfactual notation $Y(a, e)$ to be the outcome value that would be observed for a given participant in a world where we set their treatment value $A = a$ and expectancy level $E = e$.

The goal of the analysis is to formalize and answer the question ``what is the difference in the outcome variable Y (e.g., depression levels) at follow-up among those treated with psilocybin $a = 1$ or control $a = 0$, in a world where we set pre-efficacy expectancy levels, E to the same value $E = e$ across treatment arms?” This is the so-called controlled direct effect:

$$\mathbb{E}[Y(a = 1, e) - Y(a = 0, e)].$$

We show code for the simple setting where E is binary and indicates whether one does ($E = 1$) or does not ($E = 0$) expect that the treatment (that they think they received) will improve their outcome variable (e.g., Depression) at follow-up.

Possible treatment sequences

There are four possible treatment sequences (“regimes”/“policies”) of (a, e): (1,1), (1,0), (0,1), (0,0)). We can use the two pairs of these sequences to form contrasts of the causal effect of treatment when we “set” post-treatment expectancy at either $e = 0$ or $e = 1$. We have to code this as four separate functions which we do below:

```r
# A == 1, E == 0
d_A1_E0 <- function(data, trt) {
  rep( ifelse(trt == "A", 1, 0),
       nrow(data)
       )
}

# A == 0, E == 1
d_A0_E1 <- function(data, trt) {
  rep( ifelse(trt == "E", 1, 0),
       nrow(data)
  )
  }

# A == 1, E == 1
d_A1_E1 <- function(data, trt) {
  return( rep(1, nrow(data)) )
}

# A == 0, E == 0
d_A0_E0 <- function(data, trt) {
  return( rep(0, nrow(data)) )
}
```

Confounders

Now we need to include the vectors of confounders Z of the $E \rightarrow Y$ path. In the DAG in the paper, we used the vector X to denote baseline variables (pre-randomization). Here we assume that together the vector of variables (A, B, X, Z) are the complete set of common causes of $E \rightarrow Y$ path. We will distinguish between X and Z in the code below for coding convenience, but in reality, the estimation function below uses all variables (A, B, X, Z) appropriately.

First let's specify the variable names in the dataset associated with the baseline and post-treatment variables (remember these are all confounders of $E \rightarrow Y$), we only distinguish between them here for coding purposes.

```
X <- c("psych_h", "age", "mf") # baseline variables
Z <- c("SS", "B") # post-treatment confounders
```

Estimation of CDE

To target a controlled direct effect (CDE), we can pretend "A" and "E" are longitudinal treatments, which we specify in the `trt` argument. We specify the name of the outcome variable, the type (here it is continuous but this package also handles survival and binary outcome types). Since treatment A is randomized at baseline, we can specify `baseline = NULL` . In the confounders list `time_vary` , we specify that it is a list that has two elements: the first element in the list contains a vector of variable names that are confounders for $A \rightarrow Y$, and the second list element is a vector of variable names for the confounders of $E \rightarrow Y$. Because we set the variable `k = Inf` , these variables are pooled across timepoints, so the `lmtp_sdr` function treats all the variables in X and Z (concatenated together) as the confounders of the $E \rightarrow Y$ pathway. Since A is randomized, we can either set the known treatment randomization probabilities, or estimate them using the baseline variables in X. Both are valid, but here we show the case where these are estimated.

We set the `shift` variable to be the relevant treatment sequence that we set above. Remember we are interested in the two sequences where $e = 0$. `mtp` should be set to FALSE (see package documentation for details), and the participant `id` variable is set to the relevant column name in the dataset. The `folds` argument is the number of `folds` used in cross-fitting for nuisance function estimation. If the dataset is very small (e.g., n <= 75), it may make sense to set `folds = 1` . Otherwise, `folds = 2` may be prefereable. The `learners_trt` and `learners_outcome` are the machine learning prediction models involved in estimating expectancy (`learners_trt`) and the outcome (`learners_outcome`) based on their respective causes (e.g., A, X, Z). We used a GLM and a XGBoost here, but there are many options that can be used. See `SuperLearner` R package for more information on the names. Whatever learners that we use (e.g., XGBoost, random forest), make sure you install that R package first. In practice, setting 2-4 learners is probably sufficient for our purposes. We recommend including a GLM as one of them, especially since sample sizes are likely to be small-moderate in these applications, and the confounders are likely to be low dimensional (e.g., regularization with Ridge/LASSO is unlikely to be as important as in many other settings where the confounders may be high dimensional). Including some algorithms that are flexible (e.g., GAMs, Random Forest, XGBoost) may help capture non-linearities and interactions among confounders. As an example, we show how to install the packages needed for this Vignette here:

```
install.packages("lmtp")
install.packages("xgboost")
```

Low Expectancy ($e = 0$)

Now let’s fit two functions for the two combinations needed for the contrast:

$$\mathbb{E}[Y(a = 1, e = 0) - Y(a = 0, e = 0)].$$

```r
# load packages
library(lmtp)
library(xgboost)

# d_A1_E0
fit_sdr_A1E0 <- lmtp::lmtp_sdr(
  data = dat,
  trt = c("A", "E"), # second timepoint's "treatment"/exposure is expectancy, E
  outcome = "Y",
  outcome_type = "continuous",
  baseline = NULL, # no time-varying confounders: Baseline confounders are always included in Ht according to documentation
  time_vary = list(X, Z), # confounders of E -> Y path
  k = Inf, # whether to use all previous history variables pooled/concatenated
  shift = d_A1_E0,
  id = "id",
  mtp = FALSE,
  folds = 2,
  learners_trt = c("SL.glm", "SL.xgboost", "SL.mean"),
  learners_outcome = c("SL.xgboost", "SL.glm")
)

# d_A0_E0
fit_sdr_A0E0 <- lmtp::lmtp_sdr(
  data = dat,
  trt = c("A", "E"), # second timepoint's "treatment"/exposure is expectancy, E
  outcome = "Y",
  outcome_type = "continuous",
  baseline = NULL, # no time-varying confounders: Baseline confounders are always included in Ht according to documentation
  time_vary = list(X, Z), # confounders of E -> Y path
  k = Inf, # whether to use all previous history variables pooled/concatenated
  shift = d_A0_E0,
  id = "id",
  mtp = FALSE,
  folds = 2,
  learners_trt = c("SL.glm", "SL.xgboost", "SL.mean"),
  learners_outcome = c("SL.xgboost", "SL.glm")
)
```

Now that we have the two estimates above, we can calculate a point-estimate of one of the CDE contrasts, with its confidence interval and a p-value. We use the function below to compare the two fitted estimates, specifying the reference (i.e., the control and low expectancy ($a = 0$, $e = 0$) combination). We set the contrast to be on the additive scale here.

```
lmtp::lmtp_contrast(fit_sdr_A1E0, ref = fit_sdr_A0E0, type = "additive")
```

```
## LMTP Contrast: additive
```

```
## Null hypothesis: theta == 0
```

```
##
##   shift    ref estimate std.error conf.low conf.high p.value
## 1  2.33 -0.806     3.13     0.109     2.92      3.35  <0.001
```

In this dataset, we simulated the true $CDE = 3$. We can see the confidence intervals contain the true value and the point estimate is reasonably close to the truth.

High Expectancy ($e = 1$)

Now let’s estimate the mean counterfactual for the two combinations needed for the contrast:

$$\mathbb{E}[Y(a = 1, e = 1) - Y(a = 0, e = 1)].$$

We repeat these same steps with the two other relevant policies for the $e = 1$ (high expectancy) setting.

```
# E == 1
# d_A1_E1
fit_sdr_A1E1 <- lmtp::lmtp_sdr(
  data = dat,
  trt = c("A", "E"), # second timepoint's "treatment"/exposure is expectancy, E
  outcome = "Y",
  outcome_type = "continuous",
  baseline = NULL, # no time-varying confounders: Baseline confounders are always includ
ed in Ht according to documentation
  time_vary = list(X, Z), # confounders of E -> Y path
  k = Inf, # whether to use all previous history variables pooled/concatenated
  shift = d_A1_E1,
  id = "id",
  mtp = FALSE,
  folds = 2,
  learners_trt = c("SL.glm", "SL.xgboost", "SL.mean"),
  learners_outcome = c("SL.xgboost", "SL.glm")
)

# d_A0_E1
fit_sdr_A0E1 <- lmtp::lmtp_sdr(
  data = dat,
  trt = c("A", "E"), # second timepoint's "treatment"/exposure is expectancy, E
  outcome = "Y",
  outcome_type = "continuous",
  baseline = NULL, # no time-varying confounders: Baseline confounders are always includ
ed in Ht according to documentation
  time_vary = list(X, Z), # confounders of E -> Y path
  k = Inf, # whether to use all previous history variables pooled/concatenated
  shift = d_A0_E1,
  id = "id",
  mtp = FALSE,
  folds = 2,
  learners_trt = c("SL.glm", "SL.xgboost", "SL.mean"),
  learners_outcome = c("SL.xgboost", "SL.glm")
)

lmtp::lmtp_contrast(fit_sdr_A1E1, ref = fit_sdr_A0E1, type = "additive")
```

```
##
##   shift   ref estimate std.error conf.low conf.high p.value
## 1  1.67 -1.43      3.1    0.0412     3.01      3.18  <0.001
```

Again we can see the high expectancy contrast yields a value close to the true $CDE = 3$.

How to Cite

For use of this code or vignette, please cite the following papers:

Gabriel Loewinger$*$, Mats Stensrud, David Yaden, Alex Levis. A Causal Inference Framework for Effect Isolation in Randomized Studies with Unmasking.

Williams, N. and Díaz, I. (2023), “lmtp: an R package for estimating the causal effects of modified treatment policies,” Observational Studies, 9, 103–122.

Díaz, I., Williams, N., Hoffman, K. L., and Schenck, E. J. (2023), “Nonparametric causal effects based on longitudinal modified treatment policies,” Journal of the American Statistical Association, 118, 846–857.