# LMTLE: A Brief Introduction

Ashley I Naimi

April 2023

## Contents

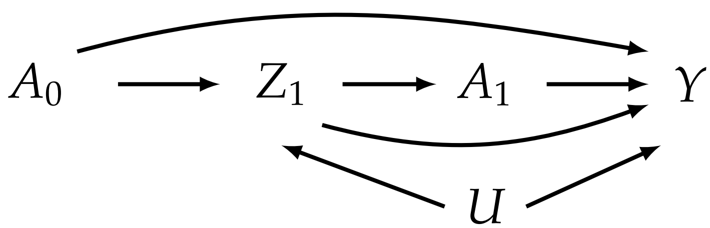
# 1 Introduction to Complex Longitudinal Data

Complex longitudinal data is becoming more common in a number of sector. These data are different from more traditional "longitudinal data" that one encounters in classical statistics courses.[1] Complex longitudinal data requires the presence of two features: first, repeated exposure, confounder, and (potentially) outcome measures; second, there has to be time-dependent feedback between these exposure, confounder, and (potentially) outcome variables.

[1] For example, courses where you would learn how to estimate models with more sophisticated correlation structures, such as generalized estimating equations or mixed effects models

This Figure demonstrates these basic conditions:

Figure 1: Causal diagram representing the structure from which the simple simulated data were generated.




This Figure is a simplified version, and several details can be added. For example, baseline covariates will always be present; there may be a $Z_0$ variable measured; there may be more than one time-dependent confounder; and the outcome may be measured at multiple time points, and may also serve as a time-dependent confounder.

Generally, the presence of a causal relation between $Z_1$ and $A_1$ suggests that $Z_1$ is a confounding variable. However, we cannot simply adjust for $Z_1$ in a standard regression model, since this would $(i)$ block part of the effect of interest from $A_0 \rightarrow Z_1 \rightarrow Y$, and $(ii)$ induce collider stratification bias through $A_0 \rightarrow Z_1 \leftarrow U \rightarrow Y$.

Because of this, we have to use specialized methods to estimate average treatment effects with complex longitudinal data.

## 2 IPW and g Computation for CLD

Perhaps the two most common techniques are inverse probability weighting or g computation (aka the parametric g formula) in such settings.

For example, if we're interested in the following ATE:

$$\psi = E(Y^{\overline{a}=1} - Y^{\overline{a}=0})$$

where $\overline{a}$ denotes the entire history of the exposure measurement from the start to the end of follow-up for each person. To make our illustrations concrete, let's use a simple simulated dataset with longitudinal information on an exposure, several time-dependent confoudners, and a time-to-event outcome.[2] Here's what the data look like:

[2] These data were simulated from Jessica Young's algorithm, modified by Erica Moodie. Details can be found in Young et al. (2010) and Moodie et al. (2014)

```
a <- read_csv(here("data", "2023_04_21-time-dependent.csv")) %>%
    group_by(ID) %>%
    mutate(exposure_lag = lag(exposure, n = 1L,
        default = 0), c1_lag = lag(c1, n = 1L,
        default = 0), c2_lag = lag(c2, n = 1L,
        default = 0)) %>%
    ungroup()

a
```

```
## # A tibble: 3,435 x 9
##       ID   Int exposure    c1    c2     Y exposure_lag c1_lag c2_lag
##    <dbl> <dbl>    <dbl> <dbl> <dbl> <dbl>        <dbl>  <dbl>  <dbl>
##  1     1     1        0     0     1     0            0      0      0
##  2     1     2        1     1     1     0            0      0      1
##  3     1     3        1     1     0     0            1      1      1
##  4     1     4        1     1     1     1            1      1      0
##  5     2     1        0     0     1     0            0      0      0
##  6     2     2        1     0     1     0            0      0      1
##  7     2     3        0     1     1     0            1      0      1
##  8     2     4        1     0     1     0            0      1      1
##  9     3     1        1     1     1     0            0      0      0
```

```
## 10     3     2         1     1     1     0             1       1       1
## # i 3,425 more rows
```

```
## look at proportion of outcome at
## each time point
a %>%
    group_by(Int) %>%
    summarise(mY = mean(Y))
```

```
## # A tibble: 4 x 2
##      Int    mY
##    <dbl> <dbl>
## 1      1 0.220
## 2      2 0.254
## 3      3 0.253
## 4      4 0.227
```

Let's talk a little about this data structure.

## 2.1 Inverse Probability Weighting

Let's start by constructing stabilized IP weights to estimate the ATE in these data:

```
# numerator
num <- glm(exposure ~ factor(Int), data = a,
    family = binomial("logit"))$fitted.values

# denominator
den <- glm(exposure ~ factor(Int) + exposure_lag +
    c1 + c2 + c1_lag + c2_lag, data = a,
    family = binomial("logit"))$fitted.values

# a_last <- a %>% mutate(last_id =
# !duplicated(ID, fromLast = T)) %>%
# filter(last_id==1) %>% select(ID,
```

```
# Int) %>% rename(last_Int = Int)
# a_last a <- left_join(a, a_last, by =
# 'ID') , cum_exp =
# cumsum(exposure)/last_Int a %>%
# select(ID, Int, exposure, Y,
# last_Int)

a <- a %>%
    mutate(sw_ = (num/den) * exposure + ((1 -
        num)/(1 - den)) * (1 - exposure)) %>%
    group_by(ID) %>%
    mutate(sw = cumprod(sw_)) %>%
    ungroup() %>%
    select(-sw_)

a %>%
    group_by(Int) %>%
    summarise(meanSW = mean(sw), maxSW = max(sw))
```

```
## # A tibble: 4 x 3
##     Int meanSW maxSW
##   <dbl>  <dbl> <dbl>
## 1     1  0.997  1.60
## 2     2  0.994  2.65
## 3     3  0.988  3.87
## 4     4  0.993  5.79
```

We can then use these weights to fit an IP weighted MSM:

```
library(lmtest)
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
```

```
##
##     as.Date, as.Date.numeric
```

```
library(sandwich)

modMSM <- lm(Y ~ factor(Int) + I(exposure/4),
    data = a, weights = sw)

coeftest(modMSM, vcov. = vcovCL(modMSM, cluster = a$ID,
    type = "HC3"))
```

```
##
## t test of coefficients:
##
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.1920508  0.0160912 11.9351  < 2e-16 ***
## factor(Int)2    0.0198237  0.0192788  1.0283  0.30390
## factor(Int)3    0.0303484  0.0222573  1.3635  0.17281
## factor(Int)4   -0.0092636  0.0241587 -0.3834  0.70141
## I(exposure/4)   0.1840595  0.0721127  2.5524  0.01074 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
install.packages("ltmle", repos = "http://lib.stat.cmu.edu/R/CRAN",
    dependencies = T)
```

```
##
## The downloaded binary packages are in
##  /var/folders/zm/rqfqp5xs0fs86qs2mcxk6q0r0000gr/T//RtmptsjYRO/downloaded_packages
```

```
library(ltmle)

# read in data again, to simplify
a <- read_csv(here("data", "2023_04_21-time-dependent.csv"))
```

```
## Rows: 3435 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (6): ID, Int, exposure, c1, c2, Y
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# convert data from long to wide
a %>%
    print(n = 3)
```

```
## # A tibble: 3,435 x 6
##      ID   Int exposure    c1    c2     Y
##   <dbl> <dbl>    <dbl> <dbl> <dbl> <dbl>
## 1     1     1        0     0     1     0
## 2     1     2        1     1     1     0
## 3     1     3        1     1     0     0
## # i 3,432 more rows
```

```
# TO DEAL WITH BASELINE CONFOUNDERS,
# KEEP THEM IN DATA
b <- a %>%
    pivot_wider(names_from = Int, values_from = c(exposure,
        c1, c2, Y)) %>%
    mutate(Y_1 = if_else(is.na(Y_1), 1, Y_1),
        Y_2 = if_else(is.na(Y_2), 1, Y_2),
        Y_3 = if_else(is.na(Y_3), 1, Y_3),
        Y_4 = if_else(is.na(Y_4), 1, Y_4)) %>%
    select(exposure_1, c1_1, c2_1, Y_1, exposure_2,
        c1_2, c2_2, Y_2, exposure_3, c1_3,
        c2_3, Y_3, exposure_4, c1_4, c2_4,
        Y_4)


b
```

```
## # A tibble: 1,228 x 16
##    exposure_1  c1_1  c2_1   Y_1 exposure_2  c1_2  c2_2   Y_2 exposure_3  c1_3
##         <dbl> <dbl> <dbl> <dbl>      <dbl> <dbl> <dbl> <dbl>      <dbl> <dbl>
##  1          0     0     1     0          1     1     1     0          1     1
##  2          0     0     1     0          1     0     1     0          0     1
##  3          1     1     1     0          1     1     1     0          1     1
##  4          0     1     1     0          0     1     1     0          1     1
##  5          1     1     1     0          1     0     1     1         NA    NA
##  6          1     1     1     0          1     1     1     1         NA    NA
##  7          1     0     1     0          0     0     1     1         NA    NA
##  8          1     1     0     0          1     1     1     0          1     1
##  9          0     0     0     0          1     1     1     0          1     1
## 10          1     1     1     1         NA    NA    NA     1         NA    NA
## # i 1,218 more rows
## # i 6 more variables: c2_3 <dbl>, Y_3 <dbl>, exposure_4 <dbl>, c1_4 <dbl>,
## #   c2_4 <dbl>, Y_4 <dbl>
```

```
# ltmle

# super learner library
sl.lib <- c("SL.mean",
            "SL.glm",
            "SL.ranger")

#ltmle
result <- ltmle(b,
                Anodes=c(paste0("exposure_",1:4)),
                Lnodes=c("c1_1", "c2_1", "c1_2", "c2_2",
                         "c1_3", "c2_3", "c1_4", "c2_4"),
                Ynodes=c("Y_1","Y_2","Y_3","Y_4"),
                survivalOutcome = TRUE,
                SL.library = list(Q = sl.lib, g = sl.lib),
                abar=list(treament = c(1, 1, 1, 1),
                          control = c(0, 0, 0, 0)),
                # estimate.time = T, need to comment out to run
```

```
                  stratify = T)
```

```
## Loading required namespace: SuperLearner

## Qform not specified, using defaults:

## formula for c1_1:

## Q.kplus1 ~ 1

## formula for c1_2:

## Q.kplus1 ~ c1_1 + c2_1

## formula for c1_3:

## Q.kplus1 ~ c1_1 + c2_1 + c1_2 + c2_2

## formula for c1_4:

## Q.kplus1 ~ c1_1 + c2_1 + c1_2 + c2_2 + c1_3 + c2_3

##

## gform not specified, using defaults:

## formula for exposure_1:

## exposure_1 ~ 1

## formula for exposure_2:

## exposure_2 ~ c1_1 + c2_1

## formula for exposure_3:

## exposure_3 ~ c1_1 + c2_1 + c1_2 + c2_2

## formula for exposure_4:

## exposure_4 ~ c1_1 + c2_1 + c1_2 + c2_2 + c1_3 + c2_3

##

## Loading required package: nnls
```

```
## Loading required namespace: ranger

## Error in pred[, "1"] : subscript out of bounds

## Timing estimate unavailable

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(result)
```

```
## Estimator:  tmle
## Call:
## ltmle(data = b, Anodes = c(paste0("exposure_", 1:4)), Lnodes = c("c1_1",
##     "c2_1", "c1_2", "c2_2", "c1_3", "c2_3", "c1_4", "c2_4"),
##     Ynodes = c("Y_1", "Y_2", "Y_3", "Y_4"), survivalOutcome = TRUE,
##     abar = list(treament = c(1, 1, 1, 1), control = c(0, 0, 0,
##         0)), stratify = T, SL.library = list(Q = sl.lib, g = sl.lib))
```

```
##
## Treatment Estimate:
##    Parameter Estimate:  0.68639
##     Estimated Std Err:  0.019482
##               p-value:  <2e-16
##     95% Conf Interval: (0.64821, 0.72458)
##
## Control Estimate:
##    Parameter Estimate:  0.4629
##     Estimated Std Err:  0.053008
##               p-value:  <2e-16
##     95% Conf Interval: (0.35901, 0.56679)
##
## Additive Treatment Effect:
##    Parameter Estimate:  0.22349
##     Estimated Std Err:  0.056474
##               p-value:  7.5755e-05
##     95% Conf Interval: (0.11281, 0.33418)
##
## Relative Risk:
##    Parameter Estimate:  1.4828
##   Est Std Err log(RR):  0.11798
##               p-value:  0.00084042
##     95% Conf Interval: (1.1767, 1.8686)
##
## Odds Ratio:
##    Parameter Estimate:  2.5396
##   Est Std Err log(OR):  0.23162
##               p-value:  5.7264e-05
##     95% Conf Interval: (1.6129, 3.9986)
```

```
result$fit$g
```

```
## [[1]]
## [[1]]$exposure_1
```

```
##             Estimate Std. Error  t value    Pr(>|t|)
## (Intercept) 0.4061438 0.05827759 6.969124 5.19302e-12
##
## [[1]]$exposure_2
##                     Risk     Coef
## SL.mean_All   0.09836547 0.379552
## SL.glm_All    0.09787365 0.620448
## SL.ranger_All 0.09776075 0.000000
##
## [[1]]$exposure_3
##                    Risk      Coef
## SL.mean_All   0.1350646 0.5519203
## SL.glm_All    0.1354099 0.4480797
## SL.ranger_All 0.1364182 0.0000000
##
## [[1]]$exposure_4
##                    Risk      Coef
## SL.mean_All   0.1172785 0.3504865
## SL.glm_All    0.1199942 0.0000000
## SL.ranger_All 0.1161943 0.6495135
##
##
## [[2]]
## [[2]]$exposure_1
##             Estimate Std. Error  t value    Pr(>|t|)
## (Intercept) 0.4061438 0.05827759 6.969124 5.19302e-12
##
## [[2]]$exposure_2
##                    Risk      Coef
## SL.mean_All   0.2353628 0.1936217
## SL.glm_All    0.2307079 0.8063783
## SL.ranger_All 0.2313254 0.0000000
##
## [[2]]$exposure_3
##                    Risk      Coef
```

```
## SL.mean_All   0.2411811 0.5080307
## SL.glm_All    0.2437590 0.1319917
## SL.ranger_All 0.2426840 0.3599776
##
## [[2]]$exposure_4
##                    Risk      Coef
## SL.mean_All   0.2727365 0.9153137
## SL.glm_All    0.3191186 0.0846863
## SL.ranger_All 0.3309709 0.0000000
```

```
result$fit$Q
```

```
## [[1]]
## [[1]]$c1_1
##              Estimate Std. Error  t value      Pr(>|t|)
## (Intercept) 0.7833164 0.03027869 25.87022 1.838207e-105
##
## [[1]]$c1_2
##                     Risk      Coef
## SL.mean_All   0.06745747 0.1493819
## SL.glm_All    0.06672632 0.8506181
## SL.ranger_All 0.16067482 0.0000000
##
## [[1]]$c1_3
##                    Risk      Coef
## SL.mean_All   0.1106053 0.4199338
## SL.glm_All    0.1100719 0.5800662
## SL.ranger_All 0.1430214 0.0000000
##
## [[1]]$c1_4
##                    Risk Coef
## SL.mean_All   0.1813361    1
## SL.glm_All    0.1951859    0
## SL.ranger_All 0.1936140    0
##
##
```

```
## [[2]]
## [[2]]$c1_1
##              Estimate Std. Error   t value    Pr(>|t|)
## (Intercept) -0.148673 0.04780011 -3.110306 0.001977839
##
## [[2]]$c1_2
##                    Risk Coef
## SL.mean_All   0.09952033    1
## SL.glm_All    0.10171704    0
## SL.ranger_All 0.12530511    0
##
## [[2]]$c1_3
##                    Risk Coef
## SL.mean_All   0.06367957    1
## SL.glm_All    0.06941183    0
## SL.ranger_All 0.07885658    0
##
## [[2]]$c1_4
##                    Risk Coef
## SL.mean_All   0.09489122    1
## SL.glm_All    0.15527797    0
## SL.ranger_All 0.11040794    0
```

## References

Erica E. M. Moodie, David A. Stephens, and Marina B. Klein. A marginal structural model for multiple-outcome survival data: assessing the impact of injection drug use on several causes of death in the canadian co-infection cohort. *Stat Med*, 33(8):1409–1425, 2014.

J. G. Young, M. A. Hernán, S. Picciotto, and J. M. Robins. Relation between three classes of structural models for the effect of a time-varying exposure on survival. *Lifetime Data Anal*, 16(1):71–84, 2010.