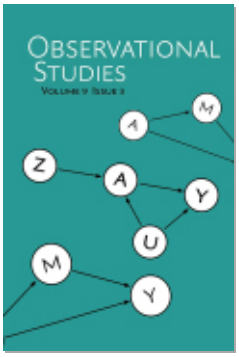
Doubly Robust Estimation of Average Treatment Effects on the Treated through Marginal Structural Models



Michael Schomaker, Philipp F. M. Baumann

Observational Studies, Volume 9, Issue 3, 2023, pp. 43-57 (Article)

Published by University of Pennsylvania Press

Observational Studies 9.3 (2023): 43-58 Published May 2023

Doubly Robust Estimation of Average Treatment Effects on the Treated through Marginal Structural Models

Michael Schomaker **michael.schomaker@stat.uni-muenchen.de**
LMU Munich
Department of Statistics
Akademiestr. 1
89799 München, Germany

Philipp F. M. Baumann **baumann@kof.ethz.ch**
ETH Zürich
KOF Swiss Economic Institute
Leonhardstrasse 21
8092 Zürich, Switzerland

Abstract

Some causal parameters are defined on subgroups of the observed data, such as the average treatment effect on the treated and variations thereof. We explain how such parameters can be defined through parameters in a marginal structural (working) model. We illustrate how existing software can be used for doubly robust effect estimation of those parameters. Our proposal for confidence interval estimation is based on the delta method. All concepts are illustrated by estimands and data from the data challenge of the *2022 American Causal Inference Conference*.

Keywords: Average Treatment Effect on the Treated, Marginal Structural Models, Data Challenge

1. Methodology and Motivation

1.1 Background and Estimands

A common estimand in causal inference is the average treatment effect on the treated (ATT); that is, the difference in expected outcomes had everyone been treated versus had everyone not been treated, for those units that actually received the treatment. This may be useful when evaluating whether an intervention actually worked among those who received it. If Y is the outcome, Z the *binary* intervention and $Y^{Z=z} = Y^z$ denotes the respective potential outcomes, then the ATT is defined as

$$\text{ATT} \quad = \quad E(Y^1|Z=1) - E(Y^0|Z=1)\,. \tag{1}$$

In some longitudinal (i.e. panel) settings, we can also define meaningful ATT's. We consider estimands, data and setups from the data challenge of the *2022 American Causal Inference Conference*. The units in the simulated data refer to patients within primary care practices. Second-layer units are the practices themselves because each practice decides whether to join the binary intervention Z, or not. In the illustrations of this paper, we

only work in this second layer where patient specific covariates (potential confounders) are available as averages for each practice. The data consist of four time points, $t = 1, 2, 3, 4$ years. We thus have covariates, the intervention and the outcome measured at multiple time points and denote those variables with X_t, Z_t and Y_t, respectively. The baseline time point is actually $t = 3$ as the intervention is only implemented as of the third year. In the given setting Z does not vary over time, i.e. $Z = 1$ means $Z_3 = 1$ and $Z_4 = 1$. We write $Z = (1, 1)$ to indicate that the intervention occurs at both $t = 3$ and $t = 4$. Broadly, relevant estimands include

$$\psi_3 = E(Y_3^1|Z = 1) - E(Y_3^0|Z = 1) \quad \text{and} \quad \psi_4 = E(Y_4^{(1,1)}|Z = 1) - E(Y_4^{(0,0)}|Z = 1)\,. \quad (2)$$

This means one is interested in the differences in the expected outcome (monthly Medicare expenditures, averaged over all patients per practice) if all practices joined the intervention versus if they did not join, both at year 3 and year 4. In the data challenge, the sample ATT is of interest, i.e. the ATT defined on the the simulated subpopulation rather than a superpopulation: we use the asterix $(*)$ to refer to this population, which may be of different size at year 3 and 4. The primary estimand is the weighted average of the two sample ATT's at year 3 and 4,

$$\psi \quad = \quad \frac{1}{N}(n_3\psi^*_{3,w} + n_4\psi^*_{4,w})\,, \quad (3)$$

where the subscript w indicates a weighted expectation, with weights given by the number of patients per practice; and n_3 and n_4 are the total number of patients in the intervention group in year 3 and year 4, respectively ($n_3 + n_4 = N$).

In addition to the usual identification assumptions of consistency, positivity and conditional exchangeability, one has to assume conditional exchangeability related to a patient selection indicator as one conditions on those patients that are available at each time point (not every patient is available in the data at each time point). We ignore this point in both the notation and discussions however. Secondary estimands relate to estimands in the spirit of (3), but conditional on subgroups defined through pre-intervention variables X_i. That is, we may replace ψ_3 with

$$\psi_{3,X_i=x_i} \quad = \quad E(Y_3^1|Z = 1, X_i = x_i) - E(Y_3^0|Z = 1, X_i = x_i)$$

and then define conditional weighted sample ATT's similar to (3).

1.2 Defining ATT's through Parameters in Marginal Structural Models

In marginal structural models (MSMs), one describes the counterfactual quantity of interest as a parametric function of intervention strategies and baseline variables, where appropriate:

$$E(Y^{Z=z}|\mathbf{X} = x) = f_\beta(Z, \mathbf{X})\,.$$

In non-saturated models, where the relationship is unknown, one may speak of "working models" to highlight the speculative nature of the assumed parametric relationship. As an example, consider the estimands of the data challenge introduced above: here, the decision whether a practice joins the intervention is made before time 3 and can thus be

considered "baseline". We denote this intervention decision with Z. At time 3 and 4 we can hypothetically intervene at each time point. We denote the intervention at these time points through $\bar{A}_t = (A_3, A_4)$. In the given example, we are interested only in $(1,1)$ and $(0,0)$ – and not $(1,0)$ and $(0,1)$. We can define a MSM as follows:

$$E(Y^{\bar{A}_t=\bar{a}_t}|\mathbf{Z}=z) \quad = \quad \beta_0 + \beta_1 A + \beta_2 Z + \beta_3 A \cdot Z\,. \tag{4}$$

where $A = 1$ if $\bar{A}_t = (1,1)$ and $A = 0$ if $\bar{A}_t = (0,0)$. The variable A is thus a so-called *summary measure* of the binary longitudinal intervention. More generally, A is a discrete variable where each category relates to a summary of a particular longitudinal intervention rule of binary variables. Based on this MSMs we can define, for instance, the following estimand:

$$E(Y^{(1,1)}|Z=1) = \beta_0 + \beta_1 \cdot 1 + \beta_2 \cdot 1 + \beta_3 \cdot 1 \cdot 1\,,$$

Similarly,

$$E(Y^{(0,0)}|Z=1) = \beta_0 + \beta_2\,.$$

The MSM thus allows us to express estimands in the spirit of (2) through the model parameters. Note that we have a saturated model because there are 4 parameters and 4 quantities defined through the two binary variables A and Z. Of course, we can more generally fit MSMs conditional on baseline covariates, time and combine the model parameters as appropriate to represent quantities of interest:

$$E(Y^{\bar{A}_t=\bar{a}_t}|\mathbf{X}=x, T=t) = f_\beta(\bar{A}_t, \mathbf{X}, t)\,. \tag{5}$$

Remark: We can use MSMs to also define the ATT of equation (1) for a single time point. Consider the ordered data $(Z, \mathbf{X}, A, Y)$: here, both Z and A are identical but we intervene only on A whereas Z is a baseline indicator defining the subgroups on which to condition on (i.e. the treated, or controls). The relevant MSM would then have to include both A and Z, as well as their interaction. This allows to define the ATT by setting $Z = 1$, setting A first as 1, then as 0 and calculating the difference. The simulation in Appendix 4.5 illustrates this point in detail and shows its validly.

Remark: In the longitudinal setup, we work with an assumed time ordering of the data of $(\mathbf{X_1}, Y_1, \mathbf{X}_2, Y_2, Z, A_3, \mathbf{X_3}, Y_3, A_4, \mathbf{X_4}, Y_4)$, see Appendix 4.1 for details.

1.3 Estimation through Targeted Maximum Likelihood Estimation

A popular approach of fitting MSMs is through inverse probability of treatment weighting (Fewell et al., 2004). However, there also exist doubly robust approaches of estimating MSMs; that is, through (longitudinal) targeted maximum likelihood estimation (LTMLE), see Petersen et al. (2014). Briefly, estimating counterfactual expected outcomes for binary longitudinal interventions (e.g., $E(Y^{\bar{a}_t})$) through longitudinal TMLE (van der Laan and Gruber, 2012) requires the evaluation of nested conditional expectations by fitting models for the (conditional) outcome, at each time point, to facilitate a standardization process with respect to time-varying confounders. Additionally, both censoring and treatment mechanisms need to be estimated at each time point to implement a targeted step that uses the

information contained in the propensity scores of the respective time points to correct the initial estimates obtained through standardization, if needed. LTMLE has the advantage over inverse probability weighted methods that it allows the incorporation of machine learning methods while retaining valid statistical inference, under assumptions (Van der Laan and Rose, 2011; Schomaker et al., 2019; Luque Fernandez et al., 2018). Fitting MSMs with LTMLE requires a refined targeted update step at each time point, estimating the counterfactual quantities for all interventions and time points of interest, stacking all the relevant counterfactual outcome vectors on top of each other, along with the respective information on intervention rules and time points to be able to generate a targeted fit of an MSM in the spirit of (5). More details can be found in Petersen et al. (2014). The approach is implemented in the *R*-package `ltmle` (Lendle et al., 2017).

Remark: There are differences between doubly robust estimation of the sample ATT and the standard ATT: while the former is not identifiable from the observed data, doubly robust approaches such as TMLE can still yield asymptotically unbiased and efficient estimates of sample parameters. We refer the interested reader to Balzer et al. (2016).

Point Estimation

We return to the data example and show how to fit the MSM specified in (4). The data are explained in Appendix 4.1. We can use the function `ltmleMSM()` of the package `ltmle` (Lendle et al., 2017). This just requires specifying the dataset (which needs to adhere to a time-ordering), the outcome, confounder and treatment variables, as well as optionally censoring variables. Specifying the intervention through an MSM requires i) the specification of the MSM itself under the option `working.msm`, ii) the actual interventions under `regimes`, as well as iii) an array that connects the specified interventions to information about time and summary measures of the binary longitudinal interventions. In our case, the summary measure is simply a binary indicator for the two interventions of interest (see Bell-Gorrod et al. (2020), Petersen et al. (2014) and the references therein for more sophisticated examples). The below box shows the code of fitting the MSM.

```
# Interventions of interest: (1,1) and (0,0)
regimesList <- list(function(row) c(1,1),
                    function(row) c(0,0))

# Defining 'summary measures' for the two interventions and 'time'
my.sum.measures <- array(c(c(1,0),c(1,1),
                           c(1,0),c(2,2))
    ,dim=c(2,2,2),dimnames=list(NULL,c("A","time"),NULL))

# Estimating MSM with ltmleMSM()
# Note: Lnodes and learning libraries have already been defined
# (see GitHub repo)
m_1 <- ltmleMSM(dwide,
                    Anodes=c("A.3","A.4"),
                    Lnodes=L_nodes,
                    Ynodes=c("Y.3","Y.4"), survivalOutcome=F,
                    Qform=NULL, gform=NULL, stratify=FALSE,
                    SL.library=ll, variance.method="tmle",
                    final.Ynodes=c("Y.3","Y.4"),
                    regimes=regimesList,
```

```
                        working.msm="Y ~ A * Z",
                        summary.measures=my.sum.measures,
                        observation.weights=
                            (dwide$n.patients.3+dwide$n.patients.4)/2
        )

# Getting estimate of (S)ATT through parameter combination
# Note: Retransformation is needed as continuous outcomes
# are transformed to lie in [0,1] for quasibinomial model
a<-attr(m_1$transformOutcome,"Yrange")[1]
b<-attr(m_1$transformOutcome,"Yrange")[2]
SATT <- (invlogit(t(c(1,1,1,1))%*%m_1$beta)*(b-a)+a) -
        (invlogit(t(c(1,0,1,0))%*%m_1$beta)*(b-a)+a)
# Note: invlogit <- function (x){1/(1 + exp(-x))}
```

It is important to highlight that for continuous outcomes `ltmle` transforms the continuous outcome to lie in the interval $[0, 1]$, based on bounds defined through the minimum and maximum observed outcome values. This is done for reasons of stability, robustness, to guarantee that outcomes outside the observed range are not predicted and the that statistical model is respected – as well as practical considerations such as appropriate modeling of heavily skewed or multimodal continuous outcomes (Gruber and van der Laan, 2010). The transformation does not affect the properties of the targeted ML estimator. Thus, to obtain the final point estimates, we have not only to combine the parameters of the MSMs appropriately but also retransform the outcome on the original scale by using the logit function together with the bounds defined through the minimum and maximum observed outcome values, see bottom lines of code and section on interval estimation below.

The above code essentially estimates $\psi^*_{4,w}$ because we first set both $Z = 1$ and $A = 1$ and then calculate the difference compared to setting $Z = 1$, but $A = 0$. The same code, with a modified MSM, can be used for other estimands, e.g. those that depend on baseline covariates and time. If we wanted to estimate ATT's conditional on time and X_3, we may simply use `working.msm="Y ~ A*Z*X3*time"` and combine the estimated MSM parameters appropriately. The Github repository (see Appendix) gives more details on this.

Remark: In the above code, we use observation weights to reflect the different number of patients per practice.

Remark: Above, we essentially estimate $\psi^*_{4,w}$. To correctly estimate ψ, addding `time` to the MSM would be needed.

Remark: Instead of using the function `ltmleMSM`, we may use the function `ltmle`, see Appendix 4.4 for an illustration. This has the advantage of using easier code, but has the disadvantage that confidence intervals for the ATT can not be easily obtained and an implementation of the parameter specific influence function may be needed (Van der Laan and Rose, 2011).

Interval Estimation

Our estimands are nonlinear combinations of parameters of MSMs. In the above example, for instance, we model ψ_4 by making use of the MSM:

$$P(\tilde{y}_i = 1|\mathbf{x}_i, z_i, t) = h(\eta_i) = \frac{\exp(\eta_i)}{1+\exp(\eta_i)} = \frac{1}{1+\exp(-\eta_i)} \ .$$

where $\tilde{Y} = (Y-a)/(b-a)$ and a, b are bounds of the outcome defined through the minimum and maximum observed outcome values. The linear predictor for observation i is $\eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_3 x_{i3} = \mathbf{x}_i'\beta$, $i = 1, 2, \ldots, n$, and $\beta = (\beta_0, \beta_1, \ldots, \beta_3)'$ is a column vector of the coefficients. In the main example, $X_1 = A, X_2 = Z, X_3 = AZ$. To estimate ψ_4 we have to use the parameter combination

$$\left\{\left(\frac{\exp(\beta_0+\beta_1+\beta_2+\beta_3)}{1+\exp(\beta_0+\beta_1+\beta_2+\beta_3)}\right)\times(b-a)+a\right\} - \left\{\left(\frac{\exp(\beta_0+\beta_2)}{1+\exp(\beta_0+\beta_2)}\right)\times(b-a)+a\right\} \ .$$

One option to approximate the standard error of the estimates of the non-linear combinations of parameters is using the Delta-Method. We have written a generic function (`msm.se`, Appendix 4.3) that calculates the standard error from an `ltmleMSM` object. It requires a specification on how the MSM's parameters should be combined. This allows the construction of confidence intervals:

```
se_SATT <- msm.se(m_1,b1=c(1,1,1,1),b2=c(1,0,1,0))
# -> (beta0+beta1+beta2*beta3) - (beta0+beta2)
SATT_lower <- SATT - qnorm(0.95)*(se_SATT/sqrt(dim(dwide)[1])) # 90% CI
SATT_upper <- SATT + qnorm(0.95)*(se_SATT/sqrt(dim(dwide)[1])) # 90% CI
```

The estimated covariance matrix, used as part of applying the delta method, can either be based on standard influence curve approaches, or on a robust approach that directly targets the asymptotic variance of the efficient influence curve (Tran et al., 2018). The latter is known to be more robust, but is not available in `ltmleMSM` for *continuous* outcomes. Our function can make use of both covariance matrices, if available. More details on the delta method and influence curve can be found in the literature (Zepeda-Tello et al., 2022).

1.4 Machine Learning Approach

As described in Section 1.3, we use TMLE for estimating the parameters of the MSM. To reduce the risk of model misspecification, our data-adaptive estimation approach considers a variety of learning and screening algorithms, both for the outcome and the treatment mechanisms (at each time point). The two sets, i.e. the screening and learning algorithms, were combined based on both computational and contextual considerations. Details can be found in Appendix 4.2 and the GitHub repository, and are inspired by the experiences of previous analyses (Baumann et al., 2021, Gehringer et al., 2018). Each screening-learning pair served as a candidate for the super learner (Van der Laan et al., 2007).

Prior to estimation, we transformed all $\mathbf{X}$ variables (see Appendix 4.1 for details on those variables) by means of the natural logarithm to stabilize the variance and higher moments of the confounders' distribution. This facilitated the stability of the numerical optimization

during the estimation as well as the robustness of the variable screening. We assessed the robustness of the variable screening by adding small perturbations to the $\mathbf{X}$ variables and could therefore ensure that the variable selection process under those perturbations was essentially identical to the variable screening without perturbations. However, we did not use these perturbations for our final model. Furthermore, we added gaussian noise variables to the set of $\mathbf{X}$ variables to check if the results were impaired, but excluded the noise in the final estimation process.

2. Data Challenge Results

2.1 Point Estimates and Bias

Figure 1 shows the distribution of our estimated ATT's across the 3400 data sets that were part of the data challenge, both for year 3 and 4 (i.e. $\hat{\psi}^*_{3,w}$ and $\hat{\psi}^*_{4,w}$). The name of our submission was `MSM+`.

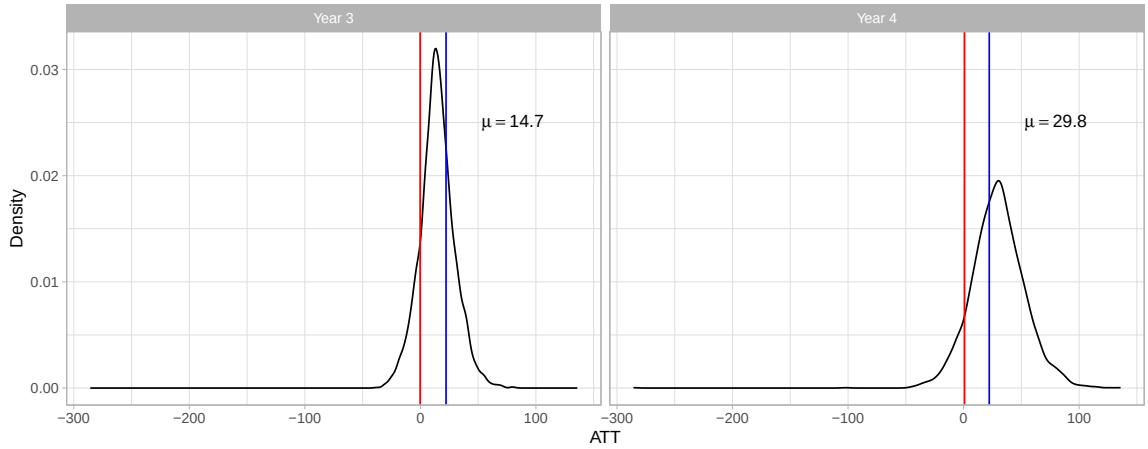


Figure 1: Kernel density estimates of the ATT distribution across 3400 estimates for the 3400 data generating processes. The blue vertical line shows the mean (μ) of the corresponding distribution. The red line indicates the mean across the 3400 true values.

It can be seen that our estimates were somewhat biased: our estimated ATTs were generally higher than the true effects. The mean absolute bias was about 15US$ for year 3 and 30US$ for year 4. Similar patterns could be observed for other estimands, i.e. estimands that condition on baseline covariates as well as ψ. This bias can be explained as follows:

1. We included time in an MSM that estimated $\psi_{3,w}$ and $\psi_{4,w}$, but not in the MSM's that estimated ψ or estimands conditional on baseline variables. We did thus not target the exactly right estimand because each model should contain time – as otherwise one cannot estimate the ATT seperately for each year and take the (weighted) average over those years, as required for most estimands of interest. It can be speculated that for the primary estimand ψ this explains up to about $(29.8\$ - 14.7\$)/2 = 7.55\$$ of the bias (under equal patient numbers in years 3 and 4). This is because we estimated $\hat{\psi}^*_{4,w}$ instead of ψ, as remarked and explained above; had we included $\hat{\psi}^*_{3,w}$ and

taken the weighted average of $\hat{\psi^*}_3$ and $\hat{\psi^*}_4$, as required by (3), then the overall mean absolute bias would have been lower, as the bias with respect to ${\psi^*}_{3,w}$ is lower (see Figure 1).

2. We used patient numbers, i.e. practice size, as observation weights when fitting the MSM. The data generating processes, which used practice sizes to define effect heterogeneity, suggests that it may have been a viable option to include those numbers additionally as covariates, if technically feasible. Another concern is that we used the average patient numbers from years 3 and 4 as weights and could not distinguish between sample sizes in different years in the required wide data format.

3. The standard `ltmleMSM` setup uses so-called "empirical weights" when fitting the MSM, which means that regimes with greater data support receive greater weight (i.e. $A = 0$ versus $A = 1$). This specific weight function does in principle affect the definition of the estimand (and should have been set to `NULL` instead).

Compared to other submissions, our average bias was neither particularly low, nor particularly high.

2.2 Coverage

The coverage of our confidence intervals was always well below the required nominal 90% coverage level. This may be explained as follows:

1. Given that our estimates were biased, it would –in any case– be unlikely to achieve nominal coverage.

2. For continuous outcomes, `ltmleMSE` can not provide the TMLE-based (robust) covariance matrix, only the standard IC-based covariance matrix. As highlighted in Section 1.3, discussed in Tran et al. (2018) and further shown in the simulation of Appendix 4.5, this is not ideal and should likely explain at least parts of the coverage performance.

2.3 Learning Algorithms

With respect to the weights for each screening-learner pair described in Appendix 4.2, two interesting patterns can be observed. First, models that focus on estimating the counterfactual outcome benefit from the full range of the provided learning algorithms. Here, the algorithms with the three highest weights are (in descending order) generalized additive models after screening with the elastic net, the CART algorithm after screening with Cramer's V, and multivariate adaptive regression splines after screening with the elastic net. Second, the final super learner for the estimating the treatment mechanisms relied exclusively on the CART algorithm (after screening with Cramer's V), giving all other algorithms a weight of 0.

3. Discussion

We have shown how to estimate ATT's using doubly robust effect estimation of marginal structural models, with integration of machine learning algorithms.

Our proposed approach has the advantage that it can be implemented easily with existing software and confidence intervals can be calculated quickly using the Delta method. However, as shown by the results of the data challenge and known from the literature, accurate interval estimation remains a major challenge in causal inference – in particular in the context of positivity violations (Li et al., 2022, Schomaker et al., 2019). It remains therefore important to evaluate and improve existing approaches to confidence intervals estimation for causal effect estimation (Tran et al., 2018).

We believe that the provided code boxes, the detailed appendix and accompanying GitHub repository make it easy to adopt our suggested approach and evaluate its performance further, beyond the specific estimands of the inspiring data challenge of the *2022 American Causal Inference Conference*.

Acknowledgments

We thank Joshua Schwab for providing many details about the functionality of the `ltmle` package and feedback on our estimation approach.

4. Appendix A.

All our code is available at the GitHub repository https://github.com/PFMB/causalchall.

4.1 Setup of Data Set

We transformed the data from long to wide format. X-variables are fixed baseline variables, whereas V-variables are time-varying X-variables (average patient related information per practice). Note that we have log-transformed those covariates and added additional noise variables for reasons explained in Section 1.4. Our time ordering is $A_t \to \mathbf{X}_t \to Y_t$ because each practice joins the program/intervention (or not) at the beginning of a year. We omit A_0 and A_1 as they are zero for everyone by definition. We included Z at the start of the data set, but any order before the first intervention node (A_3) would have been appropriate from an estimation perspective. Below is the data relating to the first data set of the data challenge.

```
   Z X1 X2 X3 X4 X5     X6     X7    X8     X9 n.patients.1 V1_avg.1
1: 1  0  A  1  A  1 20.774 14.153 0.161 43.432          113   10.808
2: 1  0  A  0  C  0 33.566  3.285 0.557 12.722          264   11.981
3: 0  0  C  1  A  1 57.283 11.178 0.257 -7.353         1309   10.950

    V2_avg.1 V3_avg.1 V4_avg.1 V5_A_avg.1  V5_B_avg.1 V5_C_avg.1       Y.1
1:     2.920    0.540    0.298      0.690       0.274      0.035 1018.8815
2:     2.947    0.530   -0.116      0.716       0.159      0.125  827.5277
3:     3.045    0.588    0.341      0.592       0.312      0.095 1070.8444

   n.patients.2 V1_avg.2 V2_avg.2 V3_avg.2 V4_avg.2 V5_A_avg.2 V5_B_avg.2
1:          109   10.768    2.872    0.532    0.274      0.706      0.266
2:          257   11.988    2.899    0.525   -0.113      0.732      0.144
3:         1342   11.034    3.061    0.581    0.302      0.576      0.325
```

```
   V5_C_avg.2       Y.2 A.3 n.patients.3 V1_avg.3 V2_avg.3  V3_avg.
1:      0.028 1607.1359   1         3181 3.094355 1.849242 0.6333972
2:      0.125 1021.4574   1          371 3.141995 1.847825 0.6119371
3:      0.099 1153.4875   0         1412 3.103510 1.880076 0.6549260

     V4_avg.3 V5_A_avg.3 V5_B_avg.3 V5_C_avg.3        Y.3 A.4 n.patients.4
1:  0.8804563  0.7124595  0.2255407 0.04018179   1245.213   1          197
2:  0.7347689  0.7104958  0.1680536 0.10975086   1123.487   1          368
3:  0.9001613  0.6413274  0.2911760 0.09349034   1298.211   0         1450

    V1_avg.4  V2_avg.4  V3_avg.4  V4_avg.4 V5_A_avg.4 V5_B_avg.4 V5_C_avg.4
1:  3.095578  1.842611 0.6450068 0.8232979  0.6941467  0.2484214 0.04114194
2:  3.160017  1.871648 0.6157260 0.6621724  0.7119689  0.1604167 0.10705907
3:  3.118923  1.896420 0.6544064 0.8394097  0.6402737  0.2844268 0.09531018


       L1_4      L2_4      L3_4       Y.4
1: 1.218347  1.404687 1.6842086 1684.8206
2: 1.433104  1.308355 1.6799247  997.6451
3: 1.154495  1.034503 1.2185981 1399.9469
```

4.2 Variable Screening and Machine Learning Algorithms

For variable screening, we used i) the LASSO with tuning parameter selection based on minimizing a generalized cross-validation criterion, ii) the elastic net (Zou and Hastie, 2005; Friedman et al., 2010) with tuning parameter selection based on the requirement to screen a fixed amount of variables (i.e., 8), iii) the random forest (Breiman, 2001; Liaw and Wiener, 2002), Cramer's V (highest association with the outcome; 4 variables selected), iv) the Pearson correlation coefficient and v) no screening. All screening algorithms could handle categorical and continuous variables simultaneously. As learning algorithms we used generalized linear models (both with main terms only and including all two-way interactions), Bayesian generalized linear models with an independent Gaussian prior distribution for the coefficients (Gelman and Su, 2021), classification and regression trees (Breiman et al., 1984; Therneau and Atkinson, 2019), multivariate adaptive (polynomial) regression splines (Friedman, 1991; Milborrow, 2021), generalized additive models (Hastie and Tibshirani, 1990; Hastie, 2020), Breimans' random forest, gradient boosting machines (Friedman, 2001; Chen et al., 2022), the arithmetic mean of the outcome/treatment and single-hidden-layer neural networks (Venables and Ripley, 2002). We further considered a multi-layer perceptron (MLP) with one to three hidden layers, regularization and normalization (Allaire and Tang, 2022) in the early stages of the analysis; however, the trade-off between predictive performance and the computational feasibility suggested dropping this algorithm in the final analysis. The two sets (i.e., the screening and learning algorithms) were combined based on both computational and contextual considerations (see GitHub repository and Baumann et al. (2021) for details). Each screening-learning pair served as a candidate for the super learner (Van der Laan et al., 2007).

4.3 Function to Generically Calculate Standard Errors with the Delta Method

```
msm.se <- function(msmobj,b1=NULL,b2=NULL,cov="IC"){

if(is.null(attr(msmobj$transformOutcome,"Yrange"))==FALSE){
    a<-attr(msmobj$transformOutcome,"Yrange")[1]
    b<-attr(msmobj$transformOutcome,"Yrange")[2]
}else{{a<-0; b<-1}}
ab <- b-a

if(is.null(b1)){stop("provide vector for b1")}
if(is.null(b2)){ind <- 0}else{ind <- 1}

comb1 <- paste(paste0(b1,"*x",1:length(b1)),collapse="+")
if(is.null(b2)==FALSE){comb2 <- paste(paste0(b2,"*x",1:length(b2)),collapse=
    "+")}else{comb2 <- 0}

quasibinform <- sprintf(paste0("~ ((exp(",comb1,"))/(1+exp(",comb1,")))* %f
    - %f *((exp(",comb2,
                                "))/(1+exp(",comb2,")))* %f"), ab, ind, ab)

if(cov=="IC"){covm <- cov(ltmle:::GetSummaryLtmleMSMInfo(msmobj,estimator="
    tmle")$IC)}else{covm <- msmobj$variance.estimate}

se <- msm::deltamethod(as.formula(quasibinform),msmobj$beta, covm);se
}
```

4.4 Estimating the ATT through `ltmle` directly

```
m_1_0 <- ltmle(dwide,
                  Anodes=c("A.3","A.4"),
                  Lnodes=c("n.patients.3", "V1_avg.3", "V2_avg.3",
                  "V3_avg.3", "V4_avg.3", "V5_A_avg.3", "V5_B_avg.3",
                  "V5_C_avg.3", "n.patients.4", "V1_avg.4", "V2_avg.4",
                  "V3_avg.4", "V4_avg.4", "V5_A_avg.4",  "V5_B_avg.4",
                  "V5_C_avg.4"),
                  Ynodes=c("Y.3","Y.4"), survivalOutcome=F,
                  SL.library=ll, variance.method="tmle"
                  abar = c(0,0)
         )

# Estimating E(Y^(0,0)|Z=1)
(mean(m_1_0$Qstar[dwide$Z == 1]))*(b-a)+a
```

4.5 Simulation

```r
# Note: the below code needs the function msm.se(), defined above
library(msm)  # for delta method (used in msm.se())
library(ltmle)

GenerateData <- function(n, abar = NULL) {
  W <- rnorm(n)
  A <- rexpit(W)
  if (is.null(abar)) {
    Y <- rexpit(W + A)
  } else {
    Y <- rexpit(W + abar)
  }

  if (is.null(abar)) {
    # observed data
    return(data.frame(W, A, Y))
  } else {
    # counterfactual mean
    return(mean(Y[A == 1])) #among treated
    # return(mean(Y))       #among all
  }
}

rexpit <- ltmle:::rexpit
invlogit <- function (x) {  1/(1 + exp(-x))}
psi0 <- GenerateData(1e6, abar = 1)
print(psi0) # true ATT

# setup for ltmleMSM
regimesList <- list(function(row) c(1),
                    function(row) c(0))

my.sum.measures <- array(c(c(1,0),c(1,1))
                         ,dim=c(2,2,1),dimnames=list(NULL,c("Int","time"),
                             NULL))

niter <- 10000 # number of simulation runs
n <- 1000      # sample size
est <-  rep(NA_real_, niter); est2 <- rep(NA_real_, niter)
coverage <- coverage2 <- matrix(NA,nrow=niter,ncol=2)

for (i in 1:niter) {
  dt <- GenerateData(n)
  r <- ltmle(dt, Anodes = "A", Ynodes = "Y", estimate.time = F, abar = 1)
  est[i] <- mean(r$Qstar[dt$A == 1]) # point estimate through ltmle
  # data for ltmleMSM: add trt A additionally as 'baseline indicator Z'
  dt2 <- dt; dt2$Z <- dt$A; dt2 <- dt2[,c("Z","W","A","Y")]
  r2 <- ltmleMSM(dt2, Anodes = "A", Ynodes = "Y",
                 final.Ynodes=c("Y"),
                 Qform = c(Y="Q.kplus1 ~ W + A"),  gform = "A ~ W",
                 regimes=regimesList,
                 working.msm="Y ~ Int * Z", # 'Int' -> see summary measures
                 summary.measures=my.sum.measures,
```

```
                variance.method="tmle"
  )
  est2[i] <- invlogit(c(1,1,1,1)%*%r2$beta) # point estimate with ltmleMSM
  # coverage; 90% CI as in data challenge
  se1 <- msm.se(r2,b1=c(1,1,1,1),b2=NULL, cov="IC")
  se2 <- msm.se(r2,b1=c(1,1,1,1),b2=NULL, cov="not IC")
  coverage[i,1] <- est2[i] - ((qnorm(0.95)*se1/sqrt(n) ))
  coverage[i,2] <- est2[i] + ((qnorm(0.95)*se1/sqrt(n) ))
  coverage2[i,1] <- est2[i] - ((qnorm(0.95)*se2/sqrt(n) ))
  coverage2[i,2] <- est2[i] + ((qnorm(0.95)*se2/sqrt(n) ))
  #
}

# give ltmle() and ltmleMSM() same results? -> YES
round(summary(est-est2),digits=8)
#    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#       0       0       0       0       0       0

# approximately unbiased? -> YES
mean(psi0-est)
mean(psi0-est2)
# 0.0003247021
# 0.0003247022

# coverage -> conservative; here, better non-IC based
mean(as.numeric(coverage[,1] <= psi0 & psi0 <= coverage[,2]))
mean(as.numeric(coverage2[,1] <= psi0 & psi0 <= coverage2[,2]))
# [1] 0.9996 # coverage IC based
# [1] 0.9673 # coverage tmle based
```

References

JJ Allaire and Yuan Tang. *tensorflow: R Interface to 'TensorFlow'*, 2022. URL https://CRAN.R-project.org/package=tensorflow. R package version 2.8.0.

Laura B. Balzer, Maya L. Petersen, Mark J. van der Laan, and the SEARCH Collaboration. Targeted estimation and inference for the sample average treatment effect in trials with and without pair-matching. *Statistics in Medicine*, 35(21):3717–3732, 2016.

P. Baumann, M. Schomaker, and E. Rossi. Estimating the effect of central bank independence on inflation using longitudinal targeted maximum likelihood estimation. *Journal of Causal Inference*, 9(1):109–146, 2021.

Helen Bell-Gorrod, Matthew P Fox, Andrew Boulle, Hans Prozesky, Robin Wood, Frank Tanser, Mary-Ann Davies, and Michael Schomaker. The impact of delayed switch to second-line antiretroviral therapy on mortality, depending on failure time definition and cd4 count at failure. *American Journal of Epidemiology*, 189:811–819, 2020.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone. *Classification and regression trees*. Routledge, 1984.

Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, Mu Li, Junyuan Xie, Min Lin, Yifeng Geng, Yutian Li, and Jiaming Yuan. *xgboost: Extreme Gradient Boosting*, 2022. URL https://CRAN.R-project.org/package=xgboost. R package version 1.5.2.1.

Z. Fewell, M. A. Hernan, F. Wolfe, K. Tilling, H. Choi, and J. A. C. Sterne. Controlling for time-dependent confounding using marginal structural models. *Stata Journal*, 4(4): 402–420, 2004.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

Jerome H Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19 (1):1–67, 1991.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

C. Gehringer, H. Rode, and M. Schomaker. The effect of electrical load shedding on pediatric hospital admissions in South Africa. *Epidemiology*, 29(6):841–847, 2018.

Andrew Gelman and Yu-Sung Su. *arm: Data Analysis Using Regression and Multilevel/Hierarchical Models*, 2021. URL https://CRAN.R-project.org/package=arm. R package version 1.12-2.

S. Gruber and M. J. van der Laan. A targeted maximum likelihood estimator of a causal effect on a bounded continuous outcome. *International Journal of Biostatistics*, 6(1): Article 26, 2010.

T.J. Hastie and R.J. Tibshirani. *Generalized Additive Models*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 1990.

Trevor Hastie. *gam: Generalized Additive Models*, 2020. URL https://CRAN.R-project.org/package=gam. R package version 1.20.

S. D. Lendle, J. Schwab, M. L. Petersen, and M. J. van der Laan. ltmle: An r package implementing targeted minimum loss-based estimation for longitudinal data. *Journal of Statistical Software*, 81(1):1–21, 2017.

Haodong Li, Sonali Rosete, Jeremy Coyle, Rachael V. Phillips, Nima S. Hejazi, Ivana Malenica, Benjamin F. Arnold, Jade Benjamin-Chung, Andrew Mertens, John M. Colford Jr, Mark J. van der Laan, and Alan E. Hubbard. Evaluating the robustness of targeted maximum likelihood estimators via realistic simulations in nutrition intervention trials. *Statistics in Medicine*, 41(12):2132–2165, 2022.

Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002. URL https://CRAN.R-project.org/doc/Rnews/.

M. A. Luque Fernandez, M. Schomaker, B. Rachet, and M. E. Schnitzer. Targeted maximum likelihood estimation for a binary treatment: A tutorial. *Statistics in Medicine*, 37:2530–2546, 2018.

Stephen Milborrow. *earth: Multivariate Adaptive Regression Splines*, 2021. URL https://CRAN.R-project.org/package=earth. R package version 5.3.1.

M. Petersen, J. Schwab, S. Gruber, N. Blaser, M. Schomaker, and M. van der Laan. Targeted maximum likelihood estimation for dynamic and static longitudinal marginal structural working models. *Journal of Causal Inference*, 2(2):147–185, 2014.

M. Schomaker, M. A. Luque Fernandez, V. Leroy, and M. A. Davies. Using longitudinal targeted maximum likelihood estimation in complex settings with dynamic interventions. *Statistics in Medicine*, 38:4888–4911, 2019.

Terry Therneau and Beth Atkinson. *rpart: Recursive Partitioning and Regression Trees*, 2019. URL https://CRAN.R-project.org/package=rpart. R package version 4.1-15.

L. Tran, M. Petersen, J. Schwab, and M. Van der Laan. Robust variance estimation and inference for causal effect estimation. *Arxiv eprints*, https://arxiv.org/abs/1810.03030, 2018.

M. Van der Laan and S. Rose. *Targeted Learning*. Springer, 2011.

M. J. van der Laan and S. Gruber. Targeted minimum loss based estimation of causal effects of multiple time point interventions. *International Journal of Biostatistics*, 8(1), 2012.

Mark J Van der Laan, Eric C Polley, and Alan E Hubbard. Super learner. *Statistical Applications in Genetics and Molecular Biology*, 6(1), 2007.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. URL https://www.stats.ox.ac.uk/pub/MASS4/. ISBN 0-387-95457-0.

R. Zepeda-Tello, M. Schomaker, C. Maringe, M. Smith, A. Belot, B. Rachet, M. E. Schnitzer, and M. A. Luque Fernandez. The delta-method and influence function in medical statistics: a reproducible tutorial. *ArXiv e-prints*, https://arxiv.org/abs/2206.15310, 2022.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320, 2005.