Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. doi: 10.18637/jss.v000.i00

lmtp: An R Package for Machine Learning and Non-Parametric Causal Effects for Longitudinal Studies

Nicholas Williams, MPH
Weill Cornell Medicine

Iván Díaz, PhD
Weill Cornell Medicine

Abstract

The majority of causal inference methods consider treatment effects based on counterfactual outcomes where exposure is deterministically assigned. When exposure is continuous, deterministic interventions may be irrelevant and impossible to bring about. As a solution, modified treatment policies offer a generalization that allows for the study of feasible interventions and offer a safeguard against positivity violations. The **lmtp** package implements the estimators of Díaz, Williams, Hoffman, and Schenck (2020) for estimating causal effects based on non-parametric modified treatment policies in `R`. In addition to modified treatment policies, the package can be used to estimate effects of deterministic and dynamic interventions. The methods provided can also be applied to both point-treatment and longitudinal settings, and can account for time-varying exposure, covariates, and right censoring, thereby providing a very general tool for causal inference in longitudinal studies. Additionally, two of the provided estimators are based on flexible machine learning regression algorithms, and avoid bias due to parametric model misspecification while maintaining valid statistical inference.

Keywords: Causal inference, longitudinal data, non-parametric, modified treatment policies, R.

1. Introduction

Most modern causal inference methods consider the effects of a treatment on a population mean outcome under interventions that set the treatment value deterministically. For example, the average treatment effect (ATE) considers the hypothetical difference in a population mean outcome if a dichotomous exposure was applied to all observations versus if it was applied to none. In the case of a continuous exposure, interventions that set the exposure to a static value deterministically are of little practical relevance. Furthermore, the estimation

of causal effects requires the so called positivity assumption which states that all observations have a greater than zero chance of experiencing the exposure value under consideration (Rosenbaum and Rubin 1983). This assumption is often violated when evaluating the effects of deterministic interventions, and is usually exacerbated with longitudinal data as the number of time points grows.

Modified treatment policies (MTPs) are a class of stochastic treatment regimes that can be formulated to avoid the above problems (Díaz and van der Laan 2012; Haneuse and Rotnitzky 2013). In a recent article (Díaz *et al.* 2020), we generalized the theoretical framework for MTPs to the longitudinal setting, accounting for time-varying treatment, covariates, and right-censoring of the outcome. Briefly, MTPs are hypothetical interventions where the post-intervention value of treatment can depend on the actual observed treatment level and the unit's history. As such, MTPs are useful to assess the effect of continuous treatments. For example, Haneuse and Rotnitzky (2013) assess the effect of reducing surgery time by a predetermined amount (e.g., 5 minutes) for lung cancer patients, where the reduction is carried out only for those patients for whom the intervention is feasible. Furthermore, MTPs generalize many important effect estimands, such as the effect of a dynamic treatment rule in which the treatment level is assigned only as a function of a unit's history. For example, dynamic treatment rules, a particular case of MTPs, may be used to estimate the effect of policies such as switching HIV treatment once the CD4 T-cell count goes below a predetermined threshold (Petersen, Tran, Geng, Reynolds, Kambugu, Wood, Bangsberg, Yiannoutsos, Deeks, and Martin 2014). MTPs also generalize many interesting causal effects such as the average treatment effects, the causal risk ratio, and causal odds ratio. In this article we describe how **lmtp** can be used for estimating the causal effects of MTPs, and present examples on using the software on several of the above cases.

The package **lmtp** implements four methods for estimating the effects of MTPs. Two of these estimators, a targeted minimum-loss based estimator (Laan and Rose 2011; Laan and Rubin 2006) and a sequentially doubly-robust estimator (SDR) (Buckley and James 1979; Fan and Gijbels 1994; van der Laan and Dudoit 2003; Rotnitzky, Faraggi, and Schisterman 2006; Rubin and Laan 2006; Kennedy, Ma, McHugh, and Small 2017), are multiply-robust. TMLE and SDR are implemented using cross-fitting to allow for the use of flexible machine learning regression methodology (Díaz *et al.* 2020). The package may be download from CRAN at `cran.r-project.org/package=lmtp`.

2. Notation and modified treatment policies

2.1. Data structure

In this article, we will use the notation of Díaz *et al.* (2020) with only slight modifications. Let i be the index of an observation from a data set with n total units and t be the index of time for a total number of time points τ. The observed data for observation i may be denoted as

$$Z_i = (W, L_1, A_1, L_2, A_2, ..., L_\tau, A_\tau, Y_{\tau+1}) \tag{1}$$

where W denotes baseline covariates, L_t denotes time-varying covariates, A_t denotes a vector

of exposure and/or censoring variables and Y denotes an outcome measured at the end of study follow-up. We observe n i.i.d. copies of Z with distribution P. We use $A_t = a_t$ to denote a realization of a random variable. If right-censoring exists, A_t can be adapted so that $A_t = (A_{1,t}, A_{2,t})$ where $A_{1,t}$ equals one if an observation is still in the study at time t and zero otherwise, and $A_{2,t}$ denotes the exposure at time t. We use an overbar to indicate the history of a variable up until time t. We then use $H_t = (\bar{L}_t, \bar{A}_{t-1})$ to denote the history of all variables up until just before A_t.

2.2. Modified treatment policies

We use the potential outcomes framework to define the causal effect of interest using our established data structure. We consider a hypothetical policy where $\bar{A}$ is set to a regime d defined as $A_t^d = d_t(A_t, H_t^d)$, where $H_t^d = (\bar{L}_t, \bar{A}_{t-1}^d)$, for a set of user-given regimes $d_t : t \in \{1, ..., \tau\}$. The defining characteristic that makes regime d_t a modified treatment policy is that it depends on the *natural value* of treatment $\bar{A}_t$, that is, the value that the treatment would have taken under no intervention. However, when the function d_t only depends on H_t, the LMTP reduces to the *dynamic treatment regimes* studied in the literature. Furthermore, when d_t is a constant that and does not depend on either A_t or H_t, then LMTPs reduce to the conventional static rules studied in the causal inference literature (e.g., Bang and Robins 2005; van der Laan and Gruber 2011). Below we present examples of all these interventions.

First, consider a study of the effect of physical activity on mortality in the elderly. Assume that each patient is monitored at several time points, and that a measure of physical activity such as the metabolic equivalent of task (MET) (Mendes, da Silva, Ramires, Reichert, Martins, Ferreira, and Tomasi 2018) is measured together with a number of lifestyle, health status, and demographic variables. In this setup, a natural question to ask would be "what is the effect on mortality of an intervention that increases physical activity by δ units for patients whose socioeconomic and health status allows it?" Formally, consider a longitudinal study with loss-to-follow-up. Let $A_t = (A_{1,t}, A_{2,t})$ where $A_{1,t}$ equals one if an observation is still in the study at time t and zero otherwise, and $A_{2,t}$ denote a continuous exposure at time t that can be changed through some intervention. A modified treatment policy that increases $A_{2,t}$ whenever it is feasible is then

$$d_t(a_t, h_t) = \begin{cases} (1, a_{2,t} + \delta_t) & \text{if } a_{2,t} + \delta_t \leq u_t(h_t) \\ (1, a_{2,t}) & \text{if } a_{2,t} + \delta_t > u_t(h_t) \end{cases} \tag{2}$$

where $u_t(h_t)$ defines the maximum level of physical activity allowed for a patient with characteristics h_t. Note that we also consider an intervention on $A_{1,t}$ because we are interested in a hypothetical world where there is no loss-to-follow-up. In this case the hypothetical exposure after intervention, A_t^d depends on the actually observed exposure, A_t. This is in contrast to a deterministic intervention where A_t^d would be set to some pre-specified value with probability one.

For dynamic treatment rules, consider a hypothetical longitudinal study where two different antiviral treatments are administered to HIV positive patients. Sometimes an antiviral drug works at first, until the virus develops resistance, at which point it is necessary to change the treatment regime. Assume we are interested in assessing a policy with two treatments encoded as $A_t\{0, 1\}$, and we want to assess the effect of a regime that would switch the

antiviral treatment as soon as the CD4 T cell count drops bellow 300. Let $A_t = (A_{1,t}, A_{2,t})$ where $A_{1,t}$ equals one if an observation is still in the study at time t and zero otherwise, and $A_{2,t}$ denotes the treatment arm at time t. Let L_t denote the CD4 T cell count at time t. In this case, one may decide to assess the effect of the rule

$$d_t(h_t) = \begin{cases} (1, 1 - a_{2,t-1}) & \text{if } l_t < 300 \\ (1, a_{2,t-1}) & \text{if } l_t \geq 300. \end{cases} \tag{3}$$

In contrast to the previous rule (2), the dynamic treatment rule (3) does not depend on the natural value of treatment at time t, it only depends on the history. This induces certain technicalities in the estimation procedure for true MTPs that depend on the natural value of treatment (Díaz *et al.* 2020). However, the software presented here handles both cases seamlessly.

It follows trivially from the above definitions that the average treatment effect from a cross-sectional study can be estimated using MTPs by simply letting $\tau = 1$ and contrasting two MTPs $d(A) = 1$ and $d(A) = 0$. The **lmtp** package presented in this article allows the contrast of different MTPs using differences, ratios, and odds ratios. We provide examples of each of these contrasts in §3.7 below.

In what follows we focus on estimating the the causal effect of MTP d on outcome Y, using **lmtp**, through the causal parameter

$$\theta = \mathsf{E}\{Y(\bar{A}^d)\}, \tag{4}$$

where $Y(\bar{A}^d)$ is the counterfactual outcome in a world, where possibly contrary to fact, each entry of $\bar{A}$ was modified according to the MTP d. When Y is continuous, θ is the mean population value of Y under MTP d; when Y is dichotomous, θ is the population proportion of event Y under MTP d. Similarly, when Y is the indicator of an event by end of the study, θ is defined as the cumulative incidence of Y under MTP d.

2.3. Identification

Our ability to estimate θ depends on the ability to identifying an expression for θ as a function of the data generating distribution P using only the observed data Z and not counterfactual variables Y^d. A full review of these identification assumptions is outside the scope of this article but is presented in our technical paper (Díaz *et al.* 2020). Briefly, the following standard assumptions must hold

Assumption 1 (Consistency) $\bar{A} = \bar{a} \implies Y = Y(\bar{a})$ *for all* $\bar{a} \in \text{supp}\, \bar{A}$

Assumption 2 (Exchangeability) *If* $(a_t, h_t) \in \text{supp}\{A_t, H_t\}$ *then* $(d(a_t, h_t), h_t) \in \text{supp}\{A_t, H_t\}$ *for* $t \in \{1, ..., \tau\}$

Assumption 3 (Positivity) $A_t \perp\!\!\!\perp Y(\bar{a})|H_t$ *for all* $\bar{a} \in \text{supp}\, \bar{A}$ *and* $t \in \{1, ..., \tau\}$

The consistency assumption states that the potential outcome for an observation under their observed exposure is the value of the outcome that we did actually observe. Assumption 2, the exchangeability assumption, is often also referred to as the no-unmeasured confounding

assumption; it is satisfied if all common causes of the exposure and outcome are measured and adjusted for. Of particular importance to this article is the positivity assumption which states that the distribution of the exposure under the MTP is supported in the data. Concretely, in a study with a continuous exposure and loss-to-follow-up, the positivity assumption states that if an observation with covariate history h_t and exposure a_t who was not lost-to-follow-up at time t exists then there is also an observation with covariate history h_t who was not lost-to-follow-up at time t but whose exposure was observed as $d(a_t, h_t)$ that also exists. The strength of MTPs is that they may be formulated to avoid violations of the positivity assumption, which is often an issue when working with continuous exposures and/or multiple time points.

Under these identification assumptions, the causal parameter θ is identified as follows. Set $\mathsf{m}_{\tau+1} = Y$. For $t = \tau, \ldots, 1$, recursively define

$$\mathsf{m}_t : (a_t, h_t) \mapsto \mathsf{E}\left[\mathsf{m}_{t+1}(A_{t+1}^d, H_{t+1}) \mid A_t = a_t, H_t = h_t\right]. \tag{5}$$

Then θ is identified as $\mathsf{E}[\mathsf{m}_1(A_1^d, L_1)]$.

3. Estimating modified treatment policy effects

3.1. Estimation methods

The **lmtp** package implements four estimation methods: a targeted minimum-loss based estimator (TMLE), a sequential doubly-robust estimator (SDR), an estimator based on the parametric G-formula, and an inverse probability weighted (IPW) estimator. We will only describe the use of TMLE, `lmtp_tmle`, and SDR, `lmtp_sdr`, as their use is strongly suggested over the others based on their advantageous theoretical properties which allow for machine learning regression while maintaining the ability to compute valid confidence intervals and p-values.

Targeted minimum-loss based estimation is a general framework for constructing asymptotically linear estimators leveraging machine learning, with an optimal bias-variance tradeoff for the target causal parameter (van der Laan and Rose 2011, 2018). In general, TMLE is constructed from a factorization of observed data likelihood into an outcome regression and an intervention mechanism. Using the outcome regression, an initial estimate of the target parameter is constructed and then *de-biased* by a fluctuation that depends on a function of the intervention mechanism. The sequential doubly-robust estimator is based on a unbiased transformation of the efficient influence function of the target estimand. For a thorough discussion of TMLE and SDR for static, dynamic, and modified treatment policies, we refer the reader to van der Laan and Gruber (2011); Luedtke, Sofrygin, van der Laan, and Carone (2017); Rotnitzky, Robins, and Babino (2017); Díaz *et al.* (2020).

TMLE and SDR require estimation of two nuisance parameters at each time point: an outcome mechanism and an intervention mechanism. Both TMLE and SDR are multiply-robust in that they allow certain configurations of nuisance parameters to be inconsistently estimated. Specifically, TMLE is considered $\tau + 1$-multiply in that it allows for inconsistent estimation of all the intervention mechanisms prior to any time point t, as long as all outcome mechanisms after time t are consistently estimated. SDR is 2^τ-robust in that at each time point, estimation

of either the intervention mechanism of at most one of the outcome mechanism is allowed to be inconsistent. Both TMLE and SDR are efficient when all the treatment mechanism and outcome regression are consistently estimated at a given consistency rate, but the SDR has better protection against model misspecification (see Luedtke *et al.* 2017; Rotnitzky *et al.* 2017; Díaz *et al.* 2020, for more details).

It is important to note that the SDR estimator can produce an estimate $\hat{\theta}$ outside of the bounds of the parameter space (e.g., probability estimates outside $[0, 1]$), while the TMLE guarantees that the estimate is within bounds of the parameter space. With this in mind and because for a single time-point TMLE and SDR are equally robust, we recommend use of TMLE for the case of a single time-point, while we recommend use of SDR for the longitudinal setting. All examples in this article will demonstrate use of both estimators.

3.2. Required data structure

Data is passed to **lmtp** estimators through the `data` argument. Data should be in wide format with one column per variable per time point under study (i.e., there should be one column for every variable in Z). These columns do not have to be in any specific order and the data set may contain variables that are not used in estimation. The names of treatment variables, censoring variables, baseline covariates, and time-varying covariates are specified using the `trt`, `cens`, `baseline`, and `time_vary` arguments respectively. The `trt`, `cens`, and `baseline` arguments accept character vectors and the `trt` and `cens` arguments should be ordered according to the time-ordering of the data generating mechanism. The `time_vary` argument accepts an unnamed list ordered according to the time-ordering of the model with each index containing the name of the time-varying covariates for the given time. The outcome variable is specified through the `outcome` argument.

Estimators are compatible with continuous, dichotomous and survival outcomes. In the case of a dichotomous or continuous outcome, only a single variable name should be passed to the `outcome` argument. For survival outcomes, a vector containing the names of the intermediate outcome and final outcome variables, ordered according to time, should be specified with the `outcome` argument. Dichotomous and survival outcomes should be coded using zero's and one's where one indicates the occurrence of an event and zero otherwise. If working with a survival outcome, once an observation experiences an outcome, all future outcome variables should also be coded with a one. The `outcome_type` argument should be set to `"continuous"` for continuous outcomes and `"binomial"` for dichotomous and survival outcomes. If the study is subject to loss-to-follow-up, the `cens` argument must be provided. Censoring indicators should be coded using zero's and one's where one indicates an observation is observed at the next time and zero indicates loss-to-follow-up. Once an observation's censoring status is switched to zero it cannot change back to one. Missing data before an observation is lost-to-follow-up is not allowed; a preprocessing step using multiple imputation is recommended for such variables.

The `k` argument controls a Markov assumption on the data generating mechanism. When `k = Inf`, the history H_t will be constructed using all previous time-point variables while setting `k` to any other value will restrict H_t to time-varying covariates from time $t - k - 1$ until $t - 1$. Baseline confounders are always included in H_t. The `create_node_list` function may be used to inspect how variables will be used for estimation. It is specified with the same `trt`, `baseline`, `time_vary`, and `k` arguments as **lmtp** estimators and is used internally to create a

"node list" that encodes which variables should be used at each time point of estimation. For example, consider a study with the observed data structure

$$Z = (W_1, W_2, L_{1,1}, L_{1,2}, A_1, L_{2,1}, L_{2,2}, A_2, Y_3) \tag{6}$$

We can translate this data structure to R with

```
R> baseline <- c("W_1", "W_2")
R> trt <- c("A_1", "A_2")
R> time_vary <- list(c("L_11", "L_12"),
R+                   c("L_21", "L_22"))
R> create_node_list(trt = trt, baseline = baseline,
R+                  time_vary = time_vary, tau = 2)

$trt
$trt[[1]]
[1] "W_1"  "W_2"  "L_11" "L_12" "A_1"

$trt[[2]]
[1] "W_1"  "W_2"  "L_11" "L_12" "L_21" "L_22" "A_1"  "A_2"


$outcome
$outcome[[1]]
[1] "W_1"  "W_2"  "L_11" "L_12" "A_1"

$outcome[[2]]
[1] "W_1"  "W_2"  "L_11" "L_12" "A_1"  "L_21" "L_22" "A_2"
```

A list of lists is returned with the names of the variables in H_t to be used for estimation of the outcome regression and the treatment mechanism at every time t. Notice that variables A_1 and A_2 are included as covariates for estimation of their own conditional probability. This is correct, and it is due to a particularity of our estimation procedure in which density ratios of treatment densities are estimated based on a classification trick using an auxiliary variable Λ as a pseudo outcome and the treatment as a predictor.

We now briefly describe how this density ratio estimation is done, although this process is fully automated and hidden from the software user. Specifically, the TMLE and SDR estimation methods require estimation of the ratio of the densities of A_t^d and A_t, conditional on the history H_t. This is achieved through computing the odds in a classification problem in an augmented dataset with $2n$ observations where the outcome is the auxiliary variable Λ (defined below) and the predictors are the variables A_t and H_t. In the $2n$ augmented data set, the data structure at time t is redefined as

$$(H_{\lambda,i,t}, A_{\lambda,i,t}, \Lambda_{\lambda,i} : \lambda = 0, 1; i = 1, ..., n) \tag{7}$$

where $\Lambda_{\lambda,i} = \lambda_i$ indexes duplicate values. For all duplicated observations $\lambda \in \{0, 1\}$ with the same i, $H_{\lambda,i,t}$ is the same. For $\lambda = 0$, $A_{\lambda,i,t}$ equals the observed exposure values $A_{i,t}$, whereas

for $\lambda = 1$, $A_{\lambda,i,t}$ equals the exposure values under the MTP d, namely A_t^d. The classification approach to density ratio estimation proceeds by estimating the conditional probability that $\Delta = 1$ in this dataset, and dividing it by the corresponding estimate of the conditional probability that $\Delta = 0$. An explanation of why this algorithm works may be found in our technical paper (Díaz *et al.* 2020).

3.3. Creating modified treatment policies

Modified treatment policies are specified using the `shift` argument, which accepts a user-defined function that returns a vector of exposure values modified according to the policy of interest. Shift functions should take two arguments, the first for specifying a data set and the second for specifying the current exposure variable. For example, a possible MTP may increase exposure by 2 units if the natural exposure value was below 5 units and do nothing otherwise. A shift function for this MTP would look like

```
R> function(data, trt) {
R+  (data[[trt]] < 5)*(data[[trt]] + 2) + (data[[trt]] >= 5)*data[[trt]]
R+ }
```

This framework is flexible and allows for specifying complex treatment regimes that can also depend on time and covariates. In the case of a binary exposure, two shift functions are installed with the package: `static_binary_on` which sets $A_{i,t} = 1$, and `static_binary_off` which sets $A_{i,t} = 0$.

3.4. The estimation engine

An attractive property of multiply-robust estimators is that they can incorporate flexible machine-learning algorithms for the estimation of nuisance parameters while remaining $\sqrt{n}$-consistent. The super learner algorithm is an ensemble learner than incorporates a set of candidate models through a weighted convex-combination based on cross-validation (van der Laan, Polley, and Hubbard 2007). Asymptotically, this weighted combination of models, called the meta-learner, will outperform any single one of its components.

Our package uses the implementation of the super learner provided by the **sl3** package (Coyle, Hejazi, Malenica, and Sofrygin 2020). Analysts must create **sl3** learner stacks which are then included in `lmtp_tmle` and `lmtp_sdr` calls with the `lrnrs_trt` and `lrnrs_outcome` arguments. The outcome variable type should guide users on selecting the appropriate candidate learners for use with the `lrnrs_outcome` argument. Regardless of whether an exposure is continuous, dichotomous, or categorical, the exposure mechanism is estimated using classification as discussed above, users should thus only include candidate learners capable of binary classification with the `lrnrs_trt` argument.

Candidate learners that rely on cross-validation for the tuning of hyper-parameters should support grouped data if used with `lrnrs_trt`. Because estimation of the treatment mechanism relies on the augmented $2n$ duplicated data set, duplicated observations must be put into the same fold during sample-splitting.

Users may install **sl3** from `https://github.com/tlverse/sl3`. Because **sl3** is not available for installation from a standard repository, it is not required to use **lmtp**. Instead, the

`lrnrs_trt` and `lrnrs_outcomes` arguments can be set equal to `NULL` and nuisance parameters will be estimated using a generalized linear model (GLM) with the `glm` function from the **stats** package (R Core Team 2020).

3.5. Additional arguments

Sample-splitting and cross-fitting is used with all methods to avoid certain technical conditions that may not hold for machine learning estimators (Zheng and van der Laan 2011; Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey, and Robins 2018), and the number of folds can be set with the `folds` argument; the minimum number of allowed folds is two. If data has a hierarchical structure, the `id` argument is used to indicate the name of a variable in the data set indicating unique groups. These identifiers will be used for generation of cross-validation folds and will be accounted for in standard error calculations. If a continuous outcome has known bounds, these bounds may be specified using the `bounds` argument with a length two numeric vector where the first index is the lower bound and the second index is the upper bound.

3.6. Contrasts

In addition to the MTP effect, researchers may be interested in a comparison of the MTP effect and the outcome under the observed exposures, or other treatment policies. Contrasts between different policies are implemented in the `lmtp_contrast` function. Users may specify any number of objects returned by calls to `lmtp_tmle` or `lmtp_sdr` to be compared to a single a reference value or a single reference MTP, specified using the `ref` argument. Depending on the outcome type, contrasts may be either additive (`type = "additive"`), an odds ratio (`type = "or"`), or the relative risk (`type = "rr"`)

3.7. Examples

Example 1: Longitudinal MTP with no loss-to-follow-up

We have simulated data on $n = 5000$ observations over a 5-month period. Each observation has a continuous exposure (`A_1`, `A_2`, `A_3`, `A_4`) and covariate (`L_1`, `L_2`, `L_3`, `L_4`) recorded at months one through four and a dichotomous outcome (`Y`) at month five. We assume no loss-to-follow-up and no Markov assumption. This data set is installed with the package and is stored in the object `sim_t4`.

For this example, we are interested in the effect of a longitudinal MTP where at each month an observation's exposure decreases by one only if their observed exposure wouldn't be less than one if modified. Our data structure has no baseline confounders and we will use only GLMs for estimation so the only objects we must specify are the treatment variables, the time-varying covariates, the outcome variable, and the MTP shift function.

```
R> trt <- c("A_1", "A_2", "A_3", "A_4")
R> time_vary <- list(c("L_1"), c("L_2"), c("L_3"), c("L_4"))
R> y <- "Y"
R> shift <- function(data, trt) {
R+   (data[[trt]] - 1) * (data[[trt]] - 1 >= 1) +
```

```
R+     data[[trt]] * (data[[trt]] - 1 < 1)
R+ }
R> lmtp_tmle(sim_t4, trt, y, time_vary = time_vary, shift = shift)

LMTP Estimator: TMLE
   Trt. Policy: (shift)

Population intervention effect
      Estimate: 0.2646
    Std. error: 0.019
        95% CI: (0.2274, 0.3019)

R> lmtp_sdr(sim_t4, trt, y, time_vary = time_vary, shift = shift)

LMTP Estimator: SDR
   Trt. Policy: (shift)

Population intervention effect
      Estimate: 0.2608
    Std. error: 0.021
        95% CI: (0.2196, 0.3019)
```

Example 2: Longitudinal MTP, right-censoring, and the super learner

For this example, we have a simulated dataset of $n = 1000$ observations. Data was simulated for three time points with a continuous time-varying exposure at times $t \in \{1, 2\}$ (`A1`, `A2`), a dichotomous time-varying covariate at times $t \in \{1, 2\}$ (`L1`, `L2`), and a dichotomous outcome (`Y`) at time $\tau + 1 = 3$. Loss-to-follow-up is present after time $t = 1$ so the data set contains censoring indicators (`C1`, `C2`). This data is installed with the package and is stored in the object `sim_cens`.

Suppose we are interested in the additive effect of an MTP where exposure is increased by 0.5 at every time point for all observations. Instead of using a linear model, we will estimate the outcome regression and treatment mechanism using a super learner composed of a GLM, a random forest (Wright and Ziegler 2017), and multivariate adaptive regression splines (Milborrow 2019).

```
R> trt <- c("A1", "A2")
R> cen <- c("C1", "C2")
R> time_vary <- list(c("L1"), c("L2"))
R> y <- "Y"
R> mtp <- function(data, trt) {
R+   data[[trt]] + 0.5
R+ }
R> lrnrs <- make_learner_stack(Lrnr_glm,
R+                             Lrnr_ranger,
R+                             Lrnr_earth)
```

```
R> tml <- lmtp_tmle(sim_cens, trt, y, time_vary = time_vary,
R+                  cens = cen, shift = mtp, learners_trt = lrnrs,
R+                  learners_outcome = lrnrs, folds = 3)
R> print(tml)

LMTP Estimator: TMLE
   Trt. Policy: (mtp)

Population intervention effect
      Estimate: 0.9011
    Std. error: 0.0094
        95% CI: (0.8826, 0.9196)

R> sdr <- lmtp_sdr(sim_cens, trt, y, time_vary = time_vary,
R+                 cens = cen, shift = mtp, learners_trt = lrnrs,
R+                 learners_outcome = lrnrs, folds = 3)
R> print(sdr)

LMTP Estimator: SDR
   Trt. Policy: (mtp)

Population intervention effect
      Estimate: 0.8995
    Std. error: 0.0095
        95% CI: (0.881, 0.918)
```

If loss-to-follow-up exists, we can estimate the population mean outcome under the observed exposures adjusting by informative loss-to-follow-up by specifying `shift = NULL`. This estimate can then be used as the reference value for calculating the additive effect of the MTP compared to the observed exposures.

```
R> tml_obs <- lmtp_tmle(sim_cens, trt, y, time_vary = time_vary,
R+                      cens = cen, shift = NULL, learners_trt = lrnrs,
R+                      learners_outcome = lrnrs, folds = 3)
R> lmtp_contrast(tml, ref = tml_obs)

  LMTP Contrast: additive
Null hypothesis: theta == 0

  theta shift   ref std.error conf.low conf.high p.value
1 0.104 0.901 0.797   0.00607   0.0918     0.116  <0.001

R> sdr_obs <- lmtp_sdr(sim_cens, trt, y, time_vary = time_vary,
R+                     cens = cen, shift = NULL, learners_trt = lrnrs,
R+                     learners_outcome = lrnrs, folds = 3)
R> lmtp_contrast(sdr, ref = sdr_obs)
```

```
  LMTP Contrast: additive
Null hypothesis: theta == 0

   theta shift ref std.error conf.low conf.high p.value
1 0.0998 0.899 0.8   0.00612   0.0878     0.112  <0.001
```

Example 3: Survival analysis and deterministic effects

The **lmtp** package may also be used to estimate deterministic causal effects, such as the causal relative risk. Suppose we have time-to-event data on $n = 2000$ observations with a time-invariant dichotomous exposure followed for a period of seven days. We wish to estimate the causal relative risk of experiencing the event by day seven. This data is installed with the package and is stored in the object `sim_point_surv`.

```
R> trt <- "trt"
R> baseline <- c("W1", "W2")
R> cens <- paste0("C.", 0:5)
R> y <- paste0("Y.", 1:6)
R> tml1 <- lmtp_tmle(sim_point_surv, trt, y, baseline, cens = cens,
R+                   learners_trt = lrnrs, learners_outcome = lrnrs,
R+                   shift = static_binary_on, folds = 3)
R> tml0 <- lmtp_tmle(sim_point_surv, trt, y, baseline, cens = cens,
R+                   learners_trt = lrnrs, learners_outcome = lrnrs,
R+                   shift = static_binary_off, folds = 3)
R> lmtp_contrast(tml1, ref = tml0, type = "rr")

  LMTP Contrast: relative risk
Null hypothesis: theta == 1

  theta shift   ref std.error conf.low conf.high p.value
1  1.22 0.812 0.665    0.0341     1.14      1.31  <0.001

R> sdr1 <- lmtp_sdr(sim_point_surv, trt, y, baseline, cens = cens,
R+                  learners_trt = lrnrs, learners_outcome = lrnrs,
R+                  shift = static_binary_on, folds = 3)
R> sdr0 <- lmtp_sdr(sim_point_surv, trt, y, baseline, cens = cens,
R+                  learners_trt = lrnrs, learners_outcome = lrnrs,
R+                  shift = static_binary_off, folds = 3)
R> lmtp_contrast(sdr1, ref = sdr0, type = "rr")

  LMTP Contrast: relative risk
Null hypothesis: theta == 1

  theta shift   ref std.error conf.low conf.high p.value
1  1.21 0.809 0.667    0.0338     1.14       1.3  <0.001
```

3.8. Extra features

Computation time can rapidly increase with many time points and when using the super learner. As a solution, **lmtp** can utilize parallel processing provided by the **future** package (Bengtsson 2020a). In addition, **lmtp** is compatible with the **progressr** package (Bengtsson 2020b) for producing progress bars by wrapping estimator calls in `with_progress`. For users familiar with the **broom** package (Robinson and Hayes 2020), **lmtp** contains a `tidy` method.

4. Reference Manual

4.1. `lmtp_tmle` and `lmtp_sdr`

Arguments

- `data`: A data frame in wide format.
- `trt`: A vector containing the column names of the treatment variables ordered by time.
- `outcome`: The column name of the outcome variable. In the case of time-to-event analysis, a vector containing the column names of the intermediate outcome variables and the final outcome variable ordered by time. Only numeric values are allowed. If the outcome type is binary, data should be coded as zeroes and ones.
- `baseline`: An optional vector containing the columns names of baseline covariates to be included for adjustment at every time-point.
- `time_vary`: A list the same length as the number of time-points under observation. The list should be ordered following the time ordering of the model. Each index of the list should be a vector containing the column names of the time-varying covariates at that time-point.
- `cens`: An optional vector, the same length as `time_vary`, containing the column names of censoring indicators. Must be provided if there is missingness in the outcome or if a time-to-event analysis.
- `shift`: A two argument function that specifies how treatment variables should be shifted.
- `k`: An integer controlling the Markov property of the data generating mechanism. If `k = Inf` (default) the history H_t will contain all previous time-point variables. If `k = 0` the history will only contain baseline variables and time-varying covariates at $t - 1$.
- `outcome_type`: The outcome variable type. Valid options are `"continuous"` and `"binomial"`.
- `id`: An optional column name containing cluster-level identifiers.

- `bounds`: An optional vector of length two containing the upper and lower bounds for a continuous outcome. If `NULL` the bounds will be taken as the minimum and maximum of the observed data; ignored if `outcome_type = "binomial"`
- `learners_outcome`: An optional **sl3** learner stack to be used for estimation of the outcome regression. If `NULL`, estimation will default to using a generalized linear model.
- `learners_trt`: An optional **sl3** learner stack to be used for estimation of the treatment mechanism. If `NULL`, estimation will default to using a generalized linear model.
- `folds`: The number of folds to be used for cross-fitting. The minimum number of allowed folds is two.
- `bound`: Determines that maximum and minimum values (scaled) predictions will be bounded to. The default is `1e-5`, bounding predictions between 1×10^{-5} and 0.9999.

Returns

Objects returned from calls to `lmtp_tmle` or `lmtp_sdr` will contain:

- `estimator`: The estimator used, either TMLE or SDR.
- `theta`: The estimated population MTP effect.
- `standard_error`: The estimated, influence function based, standard error of the MTP effect.
- `low`: The lower bound of the 95% confidence interval of the MTP effect.
- `high`: The lower bound of the 95% confidence interval of the MTP effect.
- `eif`: The estimated, uncentered, influence function.
- `shift`: The shift function specified with the `shift` argument.
- `outcome_reg`: An $n \times \tau + 1$ matrix contained the outcome regression predictions. The mean of the first column is used for calculating `theta`.
- `density_ratios`: An $n \times \tau$ matrix containing the estimated density ratios from estimation of the treatment mechanism.
- `weights_m`: A list the same length as the `folds` argument containing the super learner ensemble weights at each time-point for each fold of the outcome regression.
- `weights_r`: A list the same length as the `folds` argument containing the super learner ensemble weights at each time-point for each fold of the treatment mechanism estimation.
- `outcome_type`: The outcome variable type.

4.2. `lmtp_contrast`

Arguments

- `...`: One or more objects returned from calls to `lmtp_tmle` or `lmtp_sdr`.
- `ref`: Either a scalar reference value or another object returned from a call to `lmtp_tmle` or `lmtp_sdr`. `ref` will be compared to all other objects specified in the `...` argument.
- `type`: The contrast of interest. Valid options are `"additive"` (default) for the additive effect, `"rr"` for the relative risk, and `"or"` for the odds ratio. `"rr"` and `"or"` are only allowed when the outcome is dichotomous.

Returns

Objects returned from calls to `lmtp_contrast` will be a list containing:

- `type`: The contrast type specified with the `type` argument.
- `null`: The null hypothesis.
- `vals`: A data frame with the number of rows equal to the number of objects specified in the `...` argument. The data frame will contain columns for the contrast estimate (`"theta"`), standard error (`std.error`), and 95% confidence interval lower (`"conf.low"`) and upper (`"conf.high"`) bounds.
- `eifs`: The estimated, uncentered, influence functions of the contrast estimates.

4.3. `create_node_list`

Arguments

- `trt`: A vector containing the names of the treatment variables ordered by time.
- `tau`: An integer specifying the maximum time-point of the data generating mechanism.
- `time_vary`: A list the same length as the number of time-points under observation. The list should be ordered following the time ordering of the model. Each index of the list should be a vector containing the names of the time-varying covariates at that time-point.
- `baseline`: An optional vector containing the names of baseline covariates to be included for adjustment at every time-point.
- `k`: An integer controlling the Markov property of the data generating mechanism. If `k = Inf` (default) the history H_t will contain all previous time-point variables. If `k = 0` the history will only contain baseline variables and time-varying covariates at $t - 1$.

Returns

A list of length two. Each index of the list will contain a list of length τ with each index being a vector of the column names to be used for estimation at the corresponding time-point of either the outcome regression or treatment mechanism.

References

Bang H, Robins JM (2005). "Doubly robust estimation in missing data and causal inference models." *Biometrics*, **61**(4), 962–973.

Bengtsson H (2020a). ***future**: Unified Parallel and Distributed Processing in R for Everyone*. R package version 1.18.0, URL https://CRAN.R-project.org/package=future.

Bengtsson H (2020b). ***progressr**: A Inclusive, Unifying API for Progress Updates*. R package version 0.6.0, URL https://CRAN.R-project.org/package=progressr.

Buckley J, James I (1979). "Linear Regression with Censored Data." *Biometrika*, **66**(3), 429–436. ISSN 0006-3444. doi:10.2307/2335161. Publisher: [Oxford University Press, Biometrika Trust], URL https://www.jstor.org/stable/2335161.

Chernozhukov V, Chetverikov D, Demirer M, Duflo E, Hansen C, Newey W, Robins J (2018). "Double/debiased machine learning for treatment and structural parameters." *The Econometrics Journal*, **21**(1), C1–C68. ISSN 1368-423X. doi:10.1111/ectj.12097. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/ectj.12097, URL https://onlinelibrary.wiley.com/doi/abs/10.1111/ectj.12097.

Coyle JR, Hejazi NS, Malenica I, Sofrygin O (2020). ***sl3**: Pipelines for Machine Learning and Super Learning*. doi:10.5281/zenodo.1342293. R package version 1.3.8, URL https://github.com/tlverse/sl3.

Díaz I, van der Laan M (2012). "Population intervention causal effects based on stochastic interventions." *Biometrics*, **68**(2), 541–549. ISSN 1541-0420. doi:10.1111/j.1541-0420.2011.01685.x.

Díaz I, Williams N, Hoffman KL, Schenck EJ (2020). "Non-parametric causal effects based on longitudinal modified treatment policies." *arXiv:2006.01366*. ArXiv: 2006.01366 version: 2, URL http://arxiv.org/abs/2006.01366.

Fan J, Gijbels I (1994). "Censored Regression: Local Linear Approximations and Their Applications." *Journal of the American Statistical Association*, **89**(426), 560–570. ISSN 0162-1459. doi:10.2307/2290859. Publisher: [American Statistical Association, Taylor & Francis, Ltd.], URL https://www.jstor.org/stable/2290859.

Haneuse S, Rotnitzky A (2013). "Estimation of the effect of interventions that modify the received treatment." *Statistics in Medicine*, **32**(30), 5260–5277. ISSN 1097-0258. doi:10.1002/sim.5907. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sim.5907, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.5907.

Kennedy EH, Ma Z, McHugh MD, Small DS (2017). "Nonparametric methods for doubly robust estimation of continuous treatment effects." *Journal of the Royal Statistical Society. Series B, Statistical methodology*, **79**(4), 1229–1245. ISSN 1369-7412. doi:10.1111/rssb.12212. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5627792/.

Laan MJvd, Rose S (2011). *Targeted Learning: Causal Inference for Observational and Experimental Data*. Springer Series in Statistics. Springer-Verlag, New York. ISBN 978-1-4419-9781-4. doi:10.1007/978-1-4419-9782-1. URL https://www.springer.com/us/book/9781441997814.

Laan MJvd, Rubin D (2006). "Targeted Maximum Likelihood Learning." *The International Journal of Biostatistics*, **2**(1). ISSN 1557-4679. doi:10.2202/1557-4679.1043. Publisher: De Gruyter Section: The International Journal of Biostatistics, URL https://www.degruyter.com/view/journals/ijb/2/1/article-ijb.2006.2.1.1043.xml.xml.

Luedtke AR, Sofrygin O, van der Laan MJ, Carone M (2017). "Sequential double robustness in right-censored longitudinal models." *arXiv preprint arXiv:1705.02459*.

Mendes MdA, da Silva I, Ramires V, Reichert F, Martins R, Ferreira R, Tomasi E (2018). "Metabolic equivalent of task (METs) thresholds as an indicator of physical activity intensity." *PloS one*, **13**(7), e0200701.

Milborrow S (2019). ***earth**: Multivariate Adaptive Regression Splines*. R package version 5.1.2, URL https://CRAN.R-project.org/package=earth.

Petersen ML, Tran L, Geng EH, Reynolds SJ, Kambugu A, Wood R, Bangsberg DR, Yiannoutsos CT, Deeks SG, Martin JN (2014). "Delayed switch of antiretroviral therapy after virologic failure associated with elevated mortality among HIV-infected adults in Africa." *AIDS (London, England)*, **28**(14), 2097.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Robinson D, Hayes A (2020). ***broom**: Convert Statistical Analysis Objects into Tidy Tibbles*. R package version 0.5.6, URL https://CRAN.R-project.org/package=broom.

Rosenbaum PR, Rubin DB (1983). "The Central Role of the Propensity Score in Observational Studies for Causal Effects." *Biometrika*, **70**(1), 41–55. ISSN 0006-3444. doi:10.2307/2335942. Publisher: [Oxford University Press, Biometrika Trust], URL https://www.jstor.org/stable/2335942.

Rotnitzky A, Faraggi D, Schisterman E (2006). "Doubly Robust Estimation of the Area Under the Receiver-Operating Characteristic Curve in the Presence of Verification Bias." *Journal of the American Statistical Association*, **101**(475), 1276–1288. ISSN 0162-1459, 1537-274X. doi:10.1198/016214505000001339. URL http://www.tandfonline.com/doi/abs/10.1198/016214505000001339.

Rotnitzky A, Robins J, Babino L (2017). "On the multiply robust estimation of the mean of the g-functional." *arXiv preprint arXiv:1705.08582*.

Rubin D, Laan Mvd (2006). "Doubly Robust Censoring Unbiased Transformations." *U.C. Berkeley Division of Biostatistics Working Paper Series*. URL https://biostats.bepress.com/ucbbiostat/paper208.

van der Laan M, Dudoit S (2003). "Unified Cross-Validation Methodology For Selection Among Estimators and a General Cross-Validated Adaptive Epsilon-Net Estimator: Finite Sample Oracle Inequalities and Examples." *U.C. Berkeley Division of Biostatistics Working Paper Series*. URL https://biostats.bepress.com/ucbbiostat/paper130.

van der Laan MJ, Gruber S (2011). "Targeted minimum loss based estimation of an intervention specific mean outcome."

van der Laan MJ, Polley EC, Hubbard AE (2007). "Super Learner." *Statistical Applications in Genetics and Molecular Biology*, **6**(1). ISSN 1544-6115, 2194-6302. doi:10.2202/1544-6115.1309. Publisher: De Gruyter Section: Statistical Applications in Genetics and Molecular Biology, URL https://www.degruyter.com/view/journals/sagmb/6/1/article-sagmb.2007.6.1.1309.xml.xml.

van der Laan MJ, Rose S (2011). *Targeted Learning: Causal Inference for Observational and Experimental Data*. Springer, New York.

van der Laan MJ, Rose S (2018). *Targeted Learning in Data Science: Causal Inference for Complex longitudinal Studies*. Springer, New York.

Wright MN, Ziegler A (2017). "**ranger**: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R." *Journal of Statistical Software*, **77**(1), 1–17. doi:10.18637/jss.v077.i01.

Zheng W, van der Laan MJ (2011). "Cross-Validated Targeted Minimum-Loss-Based Estimation." In MJ van der Laan, S Rose (eds.), *Targeted Learning: Causal Inference for Observational and Experimental Data*, Springer Series in Statistics, pp. 459–474. Springer, New York, NY. ISBN 978-1-4419-9782-1. doi:10.1007/978-1-4419-9782-1_27. URL https://doi.org/10.1007/978-1-4419-9782-1_27.

Affiliation:

Nicholas Williams, MPH
Division of Biostatistics
Department of Population Health Sciences
Weill Cornell Medicine
402 East 67th Street, New York, NY 10065
E-mail: niw4001@med.cornell.edu

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd