Journal of Statistical Software

November 2012, Volume 51, Issue 13. *http://www.jstatsoft.org/*

tmle: An R Package for Targeted Maximum Likelihood Estimation

Susan Gruber
Harvard School of Public Health

Mark J. van der Laan
University of California, Berkeley

Abstract

Targeted maximum likelihood estimation (TMLE) is a general approach for constructing an efficient double-robust semi-parametric substitution estimator of a causal effect parameter or statistical association measure. **tmle** is a recently developed R package that implements TMLE of the effect of a binary treatment at a single point in time on an outcome of interest, controlling for user supplied covariates, including an additive treatment effect, relative risk, odds ratio, and the controlled direct effect of a binary treatment controlling for a binary intermediate variable on the pathway from treatment to the outcome. Estimation of the parameters of a marginal structural model is also available. The package allows outcome data with missingness, and experimental units that contribute repeated records of the point-treatment data structure, thereby allowing the analysis of longitudinal data structures. Relevant factors of the likelihood may be modeled or fit data-adaptively according to user specifications, or passed in from an external estimation procedure. Effect estimates, variances, p values, and 95% confidence intervals are provided by the software.

Keywords: causal inference, targeted maximum likelihood estimation, controlled direct effect, TMLE, MSM, R.

1. Introduction

Research in fields such as econometrics, biomedical research, and epidemiology often involves collecting data on a sample from a population in order to assess the population or group level effect of a treatment, exposure, or intervention on a measurable outcome of interest. Obtaining an unbiased and efficient estimate of the statistical parameter of interest necessitates accounting for potential bias introduced through model misspecification, informative treatment assignment, or missingness in the outcome data. Due to the curse of dimensionality, parametric estimation approaches are not feasible for high dimensional data without restrictive simplifying modeling assumptions. However, high dimensional data is increasingly common, for example in datasets used for longitudinal studies, comparative effectiveness research (administrative databases), and genomics. Targeted maximum likelihood estimation (TMLE) is an efficient, double robust, semi-parametric methodology that has been successfully applied in these settings (van der Laan and Rubin 2006; van der Laan, Rose, and Gruber 2009). The development of the **tmle** package for the R statistical programming environment (R Development Core Team 2012) was motivated by the growing need for a user-friendly tool for effective semi-parametric estimation. **tmle** is available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=tmle`.

TMLE can be applied across a broad range of problems to estimate statistical association and causal effect parameters. The methodology readily incorporates domain knowledge, user-specified parametric models, and optionally allows flexible data-adaptive estimation. The implementation of TMLE provided in the **tmle** package is restricted to estimating a variety of binary point treatment effect parameters. These parameters include marginal additive effects for binary treatments, relative risk, and odds ratio. The package also allows for estimation of the parameters of a user-specified marginal structural model (Robins 1997; Rosenblum and van der Laan 2010, MSM,), and for estimating a controlled direct effect (Pearl 2010b). Missingness is allowed in the outcome.

1.1. Causal inference

Causal effect estimation provides a useful context for describing TMLE methodology. The counterfactual framework discussed in Rubin (1974) frames the estimation of causal effects as a missing data problem. Suppose we are interested in assessing the marginal difference in an outcome, Y, if everyone received treatment ($A = 1$) vs. everyone not receiving treatment ($A = 0$). If we could actually measure the outcome under both scenarios for all individuals, the full data would be given as $X^{Full} = (Y_1, Y_0, W)$, where Y_1 is the counterfactual outcome corresponding to treatment ($A = 1$), Y_0 is the counterfactual outcome under no treatment ($A = 0$), and W is a vector of baseline covariates. A causal quantity of interest such as the additive causal effect, $\mathsf{E}(Y_1) - \mathsf{E}(Y_0)$, could be calculated as the average difference over all n subjects in X^{Full}, $1/n \sum_{i=1}^{n}(Y_{1i} - Y_{0i})$. Parameters of the full data shed light on questions of scientific interest, however in reality the full data can never be known. For each subject we can only observe the outcome corresponding to the actual treatment received. The unobserved counterfactual outcome is missing. Assume the observed data consists of n i.i.d. copies of $O = (W, A, Y = Y_A) \sim \mathsf{P}_0$, where P_0 is an unknown underlying probability distribution in a model space $\mathcal{M}$, that gives rise to the data, W is a vector of measured baseline covariates, A is a treatment variable, and Y is the outcome observed under treatment assignment A. The distribution of Y_a can be identified from the observed data distribution P_0 providing the following assumptions are met. Coarsening at random (CAR) is an assumption of conditional independence between treatment assignment and the full data given measured covariates. (Heitjan and Rubin 1991; Jacobsen and Keiding 1995; Gill, van der Laan, and Robins 1997; van der Laan and Robins 2003). Also known as *conditional exchangeability*, this assumes that there are no unmeasured confounders of the effect of treatment on the outcome. For this parameter, CAR is equivalent with the randomization assumption, $A \perp X^{Full} \mid W$. The second requirement for a causal interpretation is the positivity assumption that $\forall a \in \mathcal{A}, \mathsf{P}(A = a \mid W) > 0$. This assumption acknowledges that if no observations within some stratum defined by W receive treatment at level $A = a$, then the data do not provide sufficient information to compare the effect of treatment at level a with no treatment, or with treatment at some other level. Finally, there is a consistency assumption stating that the observed outcome value under the observed treatment is equal to the counterfactual outcome corresponding to the observed treatment.

Non-parametric structural equation modeling (NPSEM) provides an alternative paradigm for defining causal effect parameters (Pearl 2010a). The following system of equations expresses the knowledge about the data generating mechanism:

$$\begin{aligned} W &= f_W(U_W), \\ A &= f_A(W, U_A), \\ Y &= f_Y(W, A, U_Y), \end{aligned}$$

where U_W, U_A, and U_Y are exogenous error terms. This NPSEM allows the definition of counterfactual outcomes $Y_a = f_Y(W, a, U_Y)$, corresponding with the intervention that sets the treatment node A equal to a, and thereby the causal quantity of interest. This general formulation allows the functions f_W, f_A, f_Y to be entirely unspecified, or to respect exclusion restriction assumptions that strengthen identifiability by restricting the space of probability distributions under consideration, and even to assume parametric forms. From the NPSEM perspective the randomization assumption corresponds with assuming conditional independence of U_A and U_Y given W, with respect to the distribution of counterfactual Y_a.

The NPSEM approach and the counterfactual framework offer distinct formulations for discussing causality, yet each provides a foundation for defining causal effects as parameters of statistical distributions. With these definitions in place we turn our focus to obtaining an efficient, unbiased estimate of the statistical target parameter. Analysts using traditional regression models typically focus on estimating parameters of the model. However, defining the target parameter in a manner that is agnostic to the choice of model specification and fitting procedure can clarify the scientific question and expose assumptions behind different modeling choices. Separating the parameter definition from the estimation procedure allows for flexibility in the choice of estimation approach.

A number of methodologies have been applied to causal effect estimation, including the maximum likelihood-based G-computation estimator (Robins 1986), the inverse probability of treatment weighted (IPTW) estimator (Hernan, Brumback, and Robins 2000; Robins 2000a), the augmented IPTW estimator (Robins and Rotnitzky 2001; Robins, Rotnitzky, and van der Laan 2000b; Robins 2000b). Scharfstein, Rotnitzky, and Robins (1999) presented a doubly robust regression-based estimator for the treatment specific mean, later extended to time-dependent censoring (Bang and Robins 2005). We refer the interested reader to Porter, Gruber, van der Laan, and Sekhon (2011) for a discussion of TMLE in relation to these other estimators, to Moore and van der Laan (2009); Stitelman and van der Laan (2010); van der Laan and Gruber (2011) for applications of TMLE in longitudinal data analysis, and to Rosenblum and van der Laan (2010), for estimation of the parameters of an arbitrary marginal structural model.

1.2. Structure of the article

This article focuses on binary point treatment parameters that can be estimated using software provided in the current version of the **tmle** package (1.2.0-1). Section 2 of the paper provides background on causal effect estimation and defines several causal effect parameters commonly reported in the literature. Section 2 also introduces TMLE methodology, describes influence curve-based inference, and offers a brief introduction to marginal structural models. Section 3 discusses the implementation in the **tmle** package, including a discussion of data-adaptive estimation using the **SuperLearner** package (Polley and van der Laan 2012) and extensions to missing outcome data and controlled direct effect estimation. An application of the **tmle** program to the analysis of a publicly available dataset is provided in Section 4. Section 3.6 describes the application of TMLE to estimating the parameters of a MSM, and a comparison with the traditional inverse probability weighted approach described in Hernan *et al.* (2000). The final section of the paper discusses extensions to the methodology and the software. Section 6 provides answers to frequently asked questions (FAQs) regarding the practical application of TMLE using the software provided in the R package. Though the program is designed to estimate the effect of a dichotomous treatment, a valid method for estimating categorical treatment effects by separately estimating the marginal mean outcome under each level of treatment is described in the FAQ.

2. Targeted maximum likelihood estimation

2.1. Causal inference

Consider the additive effect of a binary treatment on a binary outcome with no missingness. This parameter is defined non-parametrically on full data X^{Full} as $\psi_0^F = \mathsf{E}(Y_1) - \mathsf{E}(Y_0)$, and identified from the observed data $O = (W, A, Y = Y_A)$ as $\Psi(\mathsf{P}_0) = \mathsf{E}[\mathsf{E}(Y \mid A = 1, W) - \mathsf{E}(Y \mid A = 0, W)]$ under the causal assumptions. Here ψ_0^F denotes the causal quantity of interest, and ψ_0 is the statistical counterpart that can be interpreted as the causal effect ψ_0^F under the appropriate causal assumptions. We note that Ψ represents a mapping from a probability distribution of O into a real number, called the target parameter mapping.

TMLE is a maximum likelihood based G-computation estimator that targets the fit of the data generating distribution towards reducing bias in the parameter of interest, generally one particular low-dimensional feature of the true underlying distribution. TMLE is more generally referred to as targeted minimum loss-based estimation. At its core, in the above application, TMLE methodology involves fluctuating an initial estimate of the conditional mean outcome, and minimizing a loss function to select the magnitude of the fluctuation. The targeting fluctuation is parameter-specific. The loss function is not unique, and must be chosen with care to ensure that the fluctuated estimate is a parametric sub-model $M \in \mathcal{M}$, and that the risk of the loss function is indeed minimized at the truth. Targeted *maximum likelihood* estimation corresponds with choosing the negative log-likelihood loss function. Because TMLEs solve the efficient influence curve estimating equation, and the efficient influence curves satisfies a so called double robustness property, TMLEs are guaranteed to be asymptotically unbiased if either Q_0 or g_0 is consistently estimated. When both are consistently estimated, TMLEs achieve the semi-parametric efficiency bound, under appropriate regularity conditions (van der Laan and Rubin 2006). In practice the use of a double robust estimator provides insurance against model misspecification. Since the degree to which model misspecification biases the estimate of the target parameter is never known in practice, using a double robust estimator is prudent (Neugebauer and van der Laan 2005).

An orthogonal factorization of the likelihood of the data is given by

$$\mathcal{L}(O) \quad = \quad \mathsf{P}(Y \mid A, W)\mathsf{P}(A \mid W)\mathsf{P}(W).$$

We refer to $\mathsf{P}(W)$ and $\mathsf{P}(Y \mid A, W)$ as the Q portion of the likelihood, $Q = (Q_W, Q_Y)$, and $\mathsf{P}(A \mid W)$ as the g portion of the likelihood. Further define

$$\begin{aligned} \bar{Q}_0(A, W) &\equiv \mathsf{E}(Y \mid A, W), \\ g_0(1 \mid W) &\equiv \mathsf{P}_0(A = 1 \mid W), \end{aligned}$$

where the subscript '0' denotes the truth, and a subscript 'n' will denote the corresponding quantity estimated from data. $\mathsf{P}_0(W)$ is estimated by the empirical distribution on W, the non-parametric MLE. $\bar{Q}_n(A, W)$ can be obtained by regressing Y on A and W. For some applications g_0 may be known, (e.g., treatment assignment in randomized controlled trials), so that consistent estimation will be guaranteed. It has been shown that estimation of g_0 leads to increased efficiency even when the true g_0 is known (van der Laan and Robins 2003).

The additive treatment effect, also referred to as the risk difference when the outcome is binary, is defined non-parametrically as $\mathsf{E}(Y_1) - \mathsf{E}(Y_0)$. If we let $\mu_1 = \mathsf{E}(Y_1)$ and $\mu_0 = \mathsf{E}(Y_0)$, the additive treatment effect (ATE), risk ratio (RR), and odds ratio (OR) parameters for binary outcomes are defined as:

$$\begin{aligned} \psi_0^{ATE} &= \mu_1 - \mu_0, \\ \psi_0^{RR} &= \frac{\mu_1}{\mu_0}, \\ \psi_0^{OR} &= \frac{\mu_1/(1-\mu_1)}{\mu_0/(1-\mu_0)}. \end{aligned} \tag{1}$$

Because each of these parameters is a function of (μ_0, μ_1), understanding TMLE of the parameters μ_1 and μ_0 provides a sound basis for understanding the estimation of each point treatment parameter available in the package. Notice that these parameters are functions of the Q portion of the likelihood. TMLE of a target parameter $\Psi(Q_0)$ for a specified target parameter mapping $\Psi()$ is a substitution estimator of the form $\Psi(Q_n^*)$ obtained by plugging in an estimator Q_n^* of Q_0 into the parameter mapping. The g portion of the likelihood is an ancillary nuisance parameter. If $O = (W, A, \Delta, \Delta Y_A)$, then the g-factor further factorizes into a treatment assignment mechanism, $g(A \mid W)$ and a missingness mechanism, $\pi(\Delta = 1 \mid A, W)$, where $\Delta = 1$ indicates the outcome is observed, $\Delta = 0$ indicates the outcome is missing. We will first discuss TMLE estimation when there is no missingness, then show how missingness is incorporated into the estimation procedure, and describe estimation of the population mean outcome when a subset of outcomes are unmeasured.

2.2. TMLE methodology

TMLE is a two-stage procedure. The purpose of the first stage is to get an initial estimate of the conditional mean outcome, $\bar{Q}_n^0(A, W)$. If the initial estimator of $\bar{Q}_0$ is consistent, the TMLE remains consistent, but if the initial estimator is not consistent, the subsequent targeting step provides an opportunity for TMLE to reduce any residual bias in the estimate of the parameter of interest. This is accomplished by fluctuating the initial estimate in a manner that exploits information in the g portion of the likelihood, designed to ensure that the TMLE solves the efficient influence curve estimating equation for the target parameter. Generally this is an iterative procedure, but for the ATE, RR, and OR parameters one-step convergence is mathematically guaranteed, thus $\bar{Q}_n^1(A, W) = \bar{Q}_n^*(A, W)$, where the numerical superscript denotes the kth iteration and the asterisk (*) indicates the final, targeted estimate. The idea of viewing the efficient influence curve as a path instead of an estimating equation was presented in the seminal article by van der Laan and Rubin (2006), and allows TMLE to be applied to estimate parameters where no estimating equation solution exists. This section presents the specific model for the simple case of targeting EY1 and EY0 parameters.

Given $\bar{Q}_n^0$ and g_n, fluctuating the initial density estimate is straightforward. The direction of the fluctuation determined by the efficient influence curve equations for the target parameters $\mathsf{E}(Y_1), \mathsf{E}(Y_0)$ is given by

$$H_0^*(A, W) = \frac{I(A = 0)}{g(0 \mid W)}, \tag{2}$$

$$H_1^*(A, W) = \frac{I(A = 1)}{g(1 \mid W)}. \tag{3}$$

The TMLE targeting step for updating $\bar{Q}_n^0$ with respect to $(\mathsf{E}(Y_1), \mathsf{E}(Y_0))$, is as follows:

$$\begin{aligned}
\text{logit}(\bar{Q}_n^1(A, W)) &= \text{logit}(\bar{Q}_n^0(A, W)) + \hat{\epsilon}_0 H_0^*(A, W) + \hat{\epsilon}_1 H_1^*(A, W), \\
\text{logit}(\bar{Q}_n^1(0, W)) &= \text{logit}(\bar{Q}_n^0(1, W)) + \hat{\epsilon}_0 H_0^*(0, W), \\
\text{logit}(\bar{Q}_n^1(1, W)) &= \text{logit}(\bar{Q}_n^0(0, W)) + \hat{\epsilon}_1 H_1^*(1, W).
\end{aligned}$$

The fluctuation parameter $\epsilon = (\epsilon_0, \epsilon_1)$ that controls the magnitude of the fluctuation is fit by a call to `glm`. The MLE for ϵ is obtained by a logistic regression of Y on $H_0^*(A, W), H_1^*(A, W)$, with offset $\text{logit}(Q_n^0(A, W))$. For the $\mathsf{E}(Y_1)$ and $\mathsf{E}(Y_0)$ parameters $\bar{Q}_n^*(A, W) = \bar{Q}_n^1(A, W)$.

The magnitude of $\hat{\epsilon}$ determines the degree of perturbation of the initial estimate, and is a direct function of the degree of residual confounding. For example, when $\bar{Q}_n^0$ is correct, $\hat{\epsilon}$ is essentially 0, however even this small fluctuation can reduce variance if the initial estimator of $\bar{Q}_0$ was not efficient. It is important to avoid overfitting $\bar{Q}_n^0$, as this minimizes the signal in the residuals needed for bias reduction. Section 2.4 describes how carrying out the fluctuation on the logit scale even when Y is continuous ensures that the parametric sub-model stays within the defined model space, $\mathcal{M}$.

As discussed above, estimating two parameters $\mathsf{E}(Y_1)$ and $\mathsf{E}(Y_0)$ allows us to calculate any of the causal effect parameters available for estimation in the **tmle** package. The TMLE estimate of $\mathsf{E}(Y_1)$ is given by the G-computation formula $\mathsf{E}_{W,n}(\bar{Q}_n^*(1, W)) = \frac{1}{n}\sum_{i=1}^n \bar{Q}_n^*(1, W_i)$, where the marginal distribution of W is estimated with the empirical distribution of $W_1, \ldots, W_n$. The estimate of $\mathsf{E}(Y_0)$ has an analogous definition, $\mathsf{E}_{W,n}(\bar{Q}_n^*(0, W)) = \frac{1}{n}\sum_{i=1}^n \bar{Q}_n^*(0, W_i)$. The implementation in the **tmle** package targets these two parameters simultaneously. It is also possible to target them separately, or to directly target any specific parameter. However, simultaneous targeting eliminates duplicate calculations, so is computationally sensible.

2.3. Missing outcomes

One problem that frequently arises when analyzing study data is that the outcome may not have been recorded for some observations. A naive estimation approach that considers only complete cases is inefficient, and will be biased when missingness is informative.

Causal inference parameters

Consider a randomized clinical trial measuring the effect of treatment on subsequent mortality in which a subset of people in the treatment group become ill, drop out of the study, and die shortly after being lost to follow-up. Because they are no longer in the study, outcome data is missing for these subjects. Assume that members of the treatment group who remain healthy tend to stay in the study. If observations with missing outcomes are discarded before analyzing the data the estimated effect of treatment on mortality will be overly optimistic. Thus an unbiased estimator must somehow account for this informative missingness.

TMLE does this by exploiting covariate information to reduce both bias and variance. The data are represented in a more general data structure given by $O = (W, A, \Delta, \Delta Y)$, where $\Delta = 1$ indicates the outcome is observed, $\Delta = 0$ indicates the outcome is missing, and $\Delta Y = Y$ when $\Delta = 1$, 0 otherwise. The g-factor of the likelihood now further factorizes into g_A, the treatment mechanism described above, and g_Δ, the missingness mechanism: $g_0 = \mathsf{P}(A \mid W)\mathsf{P}(\Delta \mid A, W)$. The identifiability result for $\mathsf{E}(Y_a)$ is now given by $\mathsf{E}(\bar{Q}_0(a, W))$, where $\bar{Q}_0(a, W) = \mathsf{E}(Y \mid A = a, W, \Delta = 1)$. The clever covariate for targeting the initial estimator of $\bar{Q}_0(A, W) = \mathsf{E}(Y \mid A, W, \Delta = 1)$ with respect to $\mathsf{E}(Y_a)$ is now given by $I(A = a, \Delta = 1)/g(A, \Delta \mid W)$. Thus the above clever covariates are now multiplied by $\Delta/\mathsf{P}(\Delta = 1 \mid A, W)$. The regression $\bar{Q}_0$ is estimated based on the complete observations only.

Population mean outcome

Another common research question is determining the marginal mean outcome when some observations are missing the outcome, in the absence of any treatment assignment. The data structure is given by $O = (W, \Delta, \Delta Y)$, and the only component of g is the missingness mechanism, $g_0 = \mathsf{P}(\Delta \mid W)$. The identifiability result for $\mathsf{E}(Y_1)$ is now given by $\mathsf{E}(\bar{Q}_0(W))$, where $\bar{Q}_0(W) = \mathsf{E}(Y \mid W, \Delta = 1)$. The clever covariate for this parameter is $I(\Delta = 1)/g(1 \mid W)$. The mean outcome conditional on observing the outcome is a biased estimate of the marginal mean outcome ($\mathsf{E}(Y_1)$ parameter) when missingness is informative. TMLE can reduce this bias when missingness is a function of measured baseline covariates.

2.4. Logistic loss function for continuous outcomes

One obvious approach to applying TMLE with continuous outcomes is to carry out the procedures described above on the linear scale instead of the logit scale, and indeed this has been done successfully in the past. However, particularly when there are positivity violations, this approach can lead to violations of the requirement that the fluctuation of the initial density estimate is a parametric *sub-model* of the observed data model, $\mathcal{M}$. Unlike a ogistic fluctuation, a linear fluctuation provides no assurance that the targeted estimate of the conditional mean remains within the parameter space. Gruber and van der Laan (2010b) demonstrates that the negative log likelihood for binary outcomes is a valid loss function for continuous outcomes bounded between 0 and 1, and provides a procedure for mapping outcome Y, bounded by (a, b), into Y^*, a continuous outcome bounded by (0,1): $Y^* = (Y - a)/(b - a)$. Estimates on the Y^* scale are easily mapped to their counterparts on the original scale:

$$\begin{aligned} \mathsf{E}_W(Y_0) &= \mathsf{E}_W(Y_0^*(b-a)+a), \\ \mathsf{E}_W(Y_1) &= \mathsf{E}_W(Y_1^*(b-a)+a). \end{aligned}$$

Parameter estimates $\psi_n^{ATE}, \psi_n^{RR}, \psi_n^{OR}$ are then calculated as in Equation 1.

2.5. Controlled direct effect estimation

The **tmle** package also offers controlled direct effect (CDE) estimation. Suppose that in addition to affecting outcome Y directly, treatment A gives rise to an intermediate random variable, Z, that itself has an effect on Y. For example, consider the effect of exercise, A, on weight, Y. Exercise burns calories, directly causing weight loss. Exercise may also affect caloric intake (Z), which has its own effect on weight. One research question might be, *How does weight change with daily exercise?* A second researcher might ask, *What is the effect of daily exercise on weight if caloric intake remains unchanged?* The former requires estimation of the full treatment effect of A on Y, as described above. The latter is an example of a causal effect mediated by an intermediate variable, and requires a modified estimation procedure.

The data consists of n i.i.d. copies of $O = (W, A, Z, \Delta, \Delta Y) \sim \mathsf{P}_0$, and the likelihood now factorizes as $\mathcal{L}(O) = \mathsf{P}(Y \mid \Delta = 1, Z, A, W)\mathsf{P}(\Delta = 1 \mid Z, A, W)\mathsf{P}(Z \mid A, W)\mathsf{P}(A \mid W)\mathsf{P}(W)$. Each factor can again be estimated from the data. The **tmle** package restricts controlled direct effect estimation to mediation by a binary variable, Z. Continuing the weight loss example, $Z = 0$ could indicate caloric intake is unaffected by the exercise program, while $Z = 1$ indicates increased caloric intake. CDE estimates calculated at each level of Z provide answers to the second research question posed above.

The first stage of the modified TMLE procedure estimates $\bar{Q}_0(Z, A, W)$. In the second stage $Q_n^0(Z, A, W)$ is fluctuated separately at each level of Z, using modified covariates:

$$\begin{aligned}
H_0^*(\Delta, Z, A, W) &= \frac{I(Z = z)}{g_Z(z \mid A, W)} \frac{I(A = 1)}{g_A(1 \mid W)} \frac{1}{g_\Delta(1 \mid Z, A, W)}, \\
H_1^*(\Delta, Z, A, W) &= \frac{I(Z = z)}{g_Z(z \mid A, W)} \frac{I(A = 0)}{g_A(0 \mid W)} \frac{1}{g_\Delta(1 \mid Z, A, W)}.
\end{aligned}$$

Here g_Z refers to the conditional distribution of Z given A and W, and ϵ is fit using observations where $\Delta = 1$ and $Z = z$, by default using a logistic fluctuation model.

2.6. Marginal structural models

Marginal structural models explicitly model the relationship between treatment and the marginal distribution of a treatment-specific outcome, optionally conditional on a baseline covariate vector (Robins 1997). MSMs can be applied to estimate parameters in point treatment settings as well as to longitudinal data. This discussion is restricted to point treatment models as implemented in the package. Consider estimation of a mean outcome corresponding to treatment $A = 1$, within strata defined by covariates V, modeled as $E[Y_a \mid V] = m(a, v, \psi)$. ψ can be defined as the true causal parameter, or as a statistical parameter of interest that is a projection of the true causal effect parameter onto this particular marginal structural model specification. This distinction is subtle, but important. If the MSM is misspecified, then the true MSM parameter, ψ, is not equivalent to the causal effect of interest, and any consistent estimator of ψ will necessarily not be consistent for the true causal parameter. The question of whether ψ itself is equivalent to the causal parameter is interesting and important, however from a TMLE perspective the statistical goal is to obtain an efficient unbiased estimate of ψ in the model $m(a, v, \psi)$. We therefore define the statistical target parameter as the projection onto the user-specified working MSM model, $m(a, v, \psi)$, where the projection can be weighted by a user-specified projection function of treatment and baseline covariates, $h(A, V)$.

TMLE can be applied to estimate the MSM parameter. The procedure is outlined in detail in Rosenblum and van der Laan (2010), and is described here by stepping through a simplified point treatment example. When there is no missingness in the outcome the data structure can be represented as $O = (W, A, Y)$. Let V be a subset of covariates W. Y is a continuous outcome, and we specify an MSM for the intervention-specific mean outcome under treatment set to level a as $m(a, v, \psi_0) = \mathsf{E}[Y_a|V] = \beta_0 + \beta_1 a + \beta_2 V + \beta_3 V^2$, with V univariate.

The TMLE approach to estimating $\beta = (\beta_0, \beta_1, \beta_2, \beta_3)$ rests on estimating $\bar{Q}_0$ and g_0. The first step is to obtain initial estimates of these quantities. Next the estimate $\bar{Q}_n^0$ is fluctuated in a manner designed to solve the efficient influence curve for the target parameter, β. As described above, this involves constructing a parametric sub-model that has the same dimension (d) as the number of parameters in the MSM (in our example $d = 4$). A multi-dimensional fluctuation parameter $\epsilon = (\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4)$ is fit by maximum likelihood by regressing Y on covariate $C_1(A, V)$ (defined below), with the initial estimate $\bar{Q}_n^0$ as offset (on the logit scale). The updated estimate of the conditional mean is given by $\bar{Q}_n^* = \bar{Q}_n^0 + \epsilon C_1$. C_1 is a function of the treatment assignment mechanism, the user-specified MSM, and an optional projection function that is itself a function of treatment and baseline covariates. Continuing the example, $C_1(A, V) = 1/g(A \mid V)(1, A, V, V^2)^\top$. The general form for C_1 is given in Rosenblum and van der Laan (2010). Predicted counterfactual values at each level of treatment can now be calculated for each subject. The implementation in the package is restricted to binary treatments, but this general methodology is applicable to ordinal and continuous treatments.

The final step in the algorithm is estimating β by creating a new dataset containing $2n$ observations, $O' = (W, a, \hat{Y}_a = Q_n^*(a, W))$. This new dataset contains one observation for each subject with A_i set to the value 0, and the original outcome Y replaced by $\hat{Y}_0 = Q_n^*(0, W)$, the predicted outcome when $A = 0$, and a second observation for each subject where A_i has been set to 1, and Y has been replaced by $\hat{Y}_1 = Q_n^*(1, W)$. β is estimated by regressing $\hat{Y}_a$ on the MSM with weights $= h(a, V)$. In the **tmle** package only linear or logistic regression has been implemented, but the procedure as outlined is completely general.

The inverse probability of treatment weighted (IPTW) estimation approach to estimating the parameter of an MSM is described in (Robins, Hernan, and Brumback 2000a; Hernan *et al.* 2000). In brief, this estimator weights each observation's contribution to the estimation procedure by the inverse of the conditional probability of receiving treatment given previous treatment assignment and covariate history. The parameters of the MSM are estimated using weighted regression. IPTW estimates are consistent when treatment assignment probabilities are estimated consistently. As was the case for TMLE, when the MSM is correctly specified and causal assumptions hold these estimates have a causal interpretation. In the example above, an IPTW estimate can be obtained by a weighted regression of Y on (A, V, V^2), using unstabilized weights equal to $[I(A = a)I(V = v)]/g_n(A = a \mid V)$. However, this approach is asymptotically inefficient, and can be biased for estimating ψ under misspecification of the propensity score model.

2.7. Inference

TMLE is a regular, asymptotically linear (RAL) estimator. Theory tells us that an efficient RAL estimator solves the efficient influence curve equation for the target parameter up to a second order term (Bickel, Klaassen, Ritov, and Wellner 1997). An influence curve is a function that describes the behavior of an estimator under slight perturbations of the empirical distribution (see Hampel 1974). For asymptotically linear estimators, the empirical mean of the influence curve of the estimator provides the linear approximation of the estimator. As a consequence, the variance of the influence curve provides the asymptotic variance of the estimator. Among all influence curves for RAL estimators, the one having the smallest variance is known as the efficient influence curve.

In practice, TMLE variance is estimated as the variance of the empirical influence curve divided by the number of i.i.d. units of observation, n. This quantity, $\hat{\sigma}^2$, is used to calculate p values and 95% confidence intervals. Ninety-five percent confidence intervals are calculated as $\psi_n(Q_n^*) \pm 1.96\hat{\sigma}/\sqrt{n}$ for the ATE and EY$_1$ parameters, and $\exp(\log(\psi_n(Q_n^*)) \pm 1.96\hat{\sigma}/\sqrt{n})$ for the RR and OR parameters, with $\hat{\sigma}$ equal to the estimated standard error of the $\log(RR)$ or $\log(OR)$ estimate, respectively. For CDE parameters a term reflecting the contribution of estimating Z is incorporated into each influence curve. Influence curve equations for each of the parameters estimated by the package are provided in the appendix. Variance estimates are valid when g_0 is estimated consistently.

When TMLE is applied to estimating the parameter of a marginal structural model the efficient influence curve can be used to calculate the variance-covariance matrix. The general form in the non-parametric model is given by $M^{-1}D(p)(Y,A,V,W)$, where M is a normalizing matrix, $M = -\mathsf{E}\frac{d}{d\psi}D(p)(Y,A,V,W)$, and $D(p)$ is defined as follows (see Equation 17 in Rosenblum and van der Laan 2010; Rosenblum 2011),

$$\begin{aligned} D(p)(Y,A,V,W) &= \frac{h(A,V)(Y-Q(A,V,W))}{g(A \mid V,W)}(1,A,V)^\top \\ &\quad + \sum_{a \in A} h(a,V)(Q(a,V,W)-m(a,V,\psi))\frac{\frac{d}{d\psi}m(a,v,\psi)}{m(a,v,\psi)(1-m(a,v,\psi))}. \end{aligned}$$

3. Implementation in the tmle package

The **tmle** package contains two main functions, `tmle` for estimating the ATE, RR, OR, EY1, and CDE parameters, and the `tmleMSM` function for estimating the parameter of an MSM. Each of these is discussed in turn. The TMLE algorithm is given by:

1. Obtain $\bar{Q}_n^0(A,W)$, an initial estimate of $\mathsf{P}(Y \mid A,W)$.

2. Estimate g factors needed to fluctuate $\bar{Q}_n^0(A,W)$ to obtain targeted estimate, $\bar{Q}_n^*(A,W)$.

3. Apply target parameter mapping Ψ to targeted estimate Q_n^* using the empirical distribution as estimator of the distribution of W.

The `tmle` function determines which causal effect parameter(s) to estimate based on the values of arguments specified by the user. The data arguments – `Y`, `A`, `W`, `Z`, `Delta` – are the outcome, binary treatment, baseline covariates, mediating binary variable, and missingness indicator, respectively. Only `Y` and `W` must be specified (numeric values, but there is limited support for factors). If `A` is `NULL` or has no variation (all `A` are set to 1, or all `A` are set to 0), the $\mathsf{E}(Y_1)$ parameter estimate is returned. When there is variation in `A`, the additive treatment effect is evaluated. If `Y` is binary, the RR and OR estimates are returned as well. If `Z` is not `NULL`, the parameter estimates are calculated at each level of $Z \in (0, 1)$. Each of these estimation procedures refers to `Delta` to take missingness into account, but missingness does not dictate which parameters are estimated.

When the logistic fluctuation is specified for continuous outcomes, an internal pre-processing step maps $Y \in [a, b]$ to $Y^* \in [0, 1]$ prior to calling the `estimateQ` function to carry out Step 1. `estimateQ` returns an estimate of $\bar{Q}_n^0(A, W)$ on the scale of the linear predictors needed for Step 2: the logit scale for a logistic fluctuation, linear scale for a linear fluctuation. In Step 2, the `estimateG` function is called to estimate each factor of the nuisance parameter required for calculating $H_0^*(A, W)$ and $H_1^*(A, W)$, ϵ is fit using maximum likelihood, and $\bar{Q}_n^*(A, W)$ is calculated. The `calcParameters` function estimates each parameter value, variance, p value, and constructs a 95% confidence interval. The function returns these estimates, along with values for $\bar{Q}_n^0(A, W)$, $\bar{Q}_n^*(A, W)$, and each factor of g. The package provides flexible options for estimating each relevant factor of the likelihood, allowing the procedure to be tailored to the needs of the analysis. These options and their effects are described next.

3.1. Stage 1: Estimating $\bar{Q}$

The goal of the first stage of the TMLE procedure is to fit $\bar{Q}_0$ well. A good initial fit minimizes the reliance on the targeted bias reduction step, and a target parameter estimate based on an initial fit that explains a large portion of the variance in Y generally has smaller variance than a target parameter based on a poor initial fit. TMLE achieves the semi-parametric efficiency when $\bar{Q}$ and g are both correctly specified. Several optional arguments to the `tmle` function provide flexibility in how the initial fitted values are obtained:

- `Q` $n \times 2$ matrix of fitted values for $\bar{Q}_n^0(A, W)$, $(E(Y \mid A = 0, W), E(Y \mid A = 1, W))$.
- `Qform` regression formula of the form `Y` $\sim$ `A + W`, suitable for call to **glm**.
- `Qbounds` truncation levels for Y and $\bar{Q}_n^0(A, W)$ for continuous outcomes.
- `Q.SL.library` vector of prediction algorithms for data-adaptive estimation.

Note: Estimates of $E(Y \mid Z, A, W)$ are needed for CDE parameters. These can (optionally) be supplied by passing in an $n \times 2$ matrix of predicted values $\bar{Q}_n^0(Z = 0, A, W)$ via the `Q` argument and using the `Q.Z1` argument for another $n \times 2$ matrix of predicted values $\bar{Q}_n^0(Z = 1, A, W)$ (for both arguments the first column should contain predicted values when $A = 0$, the second column when $A = 1$). `Qform` can be used to specify a regression formula that includes `A`, `W`, and `Z`.

If values are provided for more than one of these arguments, user-specified values, (`Q`, `Q.Z1`), take precedence. Data-adaptive estimation only occurs if both `Q` and `Qform` are `NULL`. The `Q` argument allows the user to incorporate any estimation procedure into `tmle` by running that procedure externally, obtaining fitted (predicted) values for each counterfactual outcome, $\bar{Q}_n^0(0, W)$ and $\bar{Q}_n^0(1, W)$ and supplying these to the `tmle` procedure. In essence, this option provides unlimited flexibility in obtaining the required stage one estimate of the conditional mean of Y.

The code snippet below shows a simple application of the `tmle` function using user-specified parametric models to estimate $\bar{Q}$ and g. First a sample of size $n = 250$ is drawn from a data generating distribution with true parameter values $\psi_0^{ATE} = 0.216, \psi_0^{RR} = 1.395, \psi_0^{OR} = 2.659$.

Baseline covariates $W = (W_1, W_2, W_3) \sim_{i.i.d.} N(0, 1)$ are simulated for each subject. These values are used to selectively assign treatment, A, and then a binary outcome that is a function of treatment and all baseline covariates is simulated.

```
R> n <- 250
R> W <- matrix(rnorm(n * 3), ncol = 3)
R> colnames(W) <- paste("W", 1:3, sep = "")
R> A <- rbinom(n, 1, plogis(0.6 * W[,1] + 0.4 * W[,2] + 0.5 * W[,3]))
R> Y <- rbinom(n, 1, plogis(A + 0.2 * W[,1] + 0.1 * W[,2] + 0.2 * W[,3]^2))
```

Next, parameters are estimated based on correctly specified models for the Q and g factors of the likelihood. The models are passed as arguments to the function, along with data arguments (Y, A, W). Default settings imply there is no missing outcome data and that observations are i.i.d.

```
R> result.Qcgc <- tmle(Y, A, W, family = "binomial",
+    Qform = Y ~ A + W1 + W2 + W3, gform = A ~ W1 + W2 + W3)
R> result.Qcgc

 Additive Effect
   Parameter Estimate:  0.21157
   Estimated Variance:  0.0044941
              p-value:  0.0015995
    95% Conf Interval: (0.080178, 0.34297)

 Relative Risk
   Parameter Estimate:  1.3966
              p-value:  0.0025233
    95% Conf Interval: (1.1244, 1.7347)

              log(RR):  0.33406
    variance(log(RR)):  0.012232

 Odds Ratio
   Parameter Estimate:  2.5554
              p-value:  0.0025418
    95% Conf Interval: (1.3895, 4.6995)

              log(OR):  0.93822
    variance(log(OR)):  0.096621
```

tmle relies on the **SuperLearner** package to provide data-adaptive estimation (Polley and van der Laan 2012). Super learning is an ensemble method that relies on proven oracle properties of V-fold cross validation to ascertain an optimal convex combination of estimates obtained from application of each algorithm in a user-specified library of prediction algorithms (van der Laan, Polley, and Hubbard 2007). Because one cannot know in advance which class of procedures will be most successful for a given problem, an important aspect of super learning is ensuring that the library of prediction algorithms includes a variety of approaches that search over a large space of possible models. For example, one might include a collection of prespecified regression models (main terms, main terms plus key interaction terms) along with other flexible modeling approaches, such as non-linear models, cubic splines, and classifiers. (Note that **tmle** version $\geq$ 1.2-0 is compatible with all versions of **SuperLearner** through 2.0-6.)

The following example applies super learning to the data generated in the first example above in order to estimate $\bar{Q}_0$. The user-specified library contains three prediction algorithms: 1) `SL.glm` is a main terms regression of Y on A and W, 2) `SL.step` calls the `step` function distributed with the base R installation (R Development Core Team 2012) with forward and backward moves incorporating quadratic terms, and 3) `SL.DSA.2` calls the `DSA` function in the suggested **DSA** package that uses deletion and addition moves to search over a space of polynomial models that is in this case constrained to order two (Neugebauer and Bullard 2010). In contrast to the AIC criterion used by the `step` procedure, `DSA` model selection is based on cross-validation (Sinisi and van der Laan 2004).

```
R> result.QSLgc <- tmle(Y, A, W, family = "binomial",
+    Q.SL.library = c("SL.glm", "SL.step", "SL.DSA.2"),
+    gform = A ~ W1 + W2 + W3)
R> summary(result.QSLgc)

 Initial estimation of Q
         Procedure: SuperLearner
         Model:
                 Y ~  SL.glm_All + SL.step_All + SL.DSA.2_All

         Coefficients:
              SL.glm_All    0
             SL.step_All    0
            SL.DSA.2_All    1

 Estimation of g (treatment mechanism)
         Procedure: user-supplied regression formula
         Model:
                 A ~  (Intercept) + W1 + W2 + W3

         Coefficients:
             (Intercept)   -0.01499195
                      W1    0.7587852
                      W2    0.2719946
                      W3    0.3438723

 Estimation of g.Z (intermediate variable assignment mechanism)
         Procedure: No intermediate variable

 Estimation of g.Delta (missingness mechanism)
         Procedure: No missingness
```

```
Bounds on g: ( 0.025 0.975 )

Additive Effect
  Parameter Estimate:  0.20889
  Estimated Variance:  0.0045076
             p-value:  0.0018622
   95% Conf Interval: (0.077302, 0.34049)

Relative Risk
  Parameter Estimate:  1.3884
             p-value:  0.0027473
   95% Conf Interval: (1.1201, 1.721)

             log(RR):  0.32814
   variance(log(RR)):  0.012006

Odds Ratio
  Parameter Estimate:  2.5336
             p-value:  0.0030238
   95% Conf Interval: (1.3705, 4.6839)

             log(OR):  0.92965
   variance(log(OR)):  0.098287
```

These parameter estimates and variances using super learning are very similar to those obtained using the correctly specified regression model for $\bar{Q}$, signaling that data-adaptive estimation was successful at closely approximating the true regression of Y on A and W. `tmle`'s default library for estimating $\bar{Q}_0$ contains three algorithms available with the base installation of R, `SL.glm, SL.step` and `GL.glm.interaction`, a glm variant that includes second order terms. However, a larger library that incorporates additional estimation procedures is recommended. If the **SuperLearner** package is not available, in the absence of a user-specified regression formula the function will fail. (In earlier versions of the package ($<$ 1.2-0) under these circumstances $\bar{Q}_0$ was estimated using a main terms regression of Y on A and W.)

The summary method for `tmle` objects lists the procedures used to estimate the relevant Q and g factors of the likelihood. The super learner is a convex combination of predicted values. When super learning is used, coefficients reported in the summary reflect each prediction algorithm's contribution. A coefficient of 0 signifies that incorporating predictions from that algorithm does not substantially improve the overall fit given the predictions from algorithms with non-zero coefficients, however, this should not be interpreted as a goodness-of-fit measure. For example, if two model selection algorithms arrive at the exact same model, at most one will have a non-zero coefficient.

It is important to avoid overfitting $\bar{Q}_n^0$, as this minimizes the signal in the residuals needed for bias reduction. The `tmle` function provides an option for guarding against overfits by cross-validating the initial super learner estimate of $\bar{Q}_0$. Independent units of observation are evenly divided among V folds. Observational units are identified by the `id` variable, an optional argument to the function that if not specified implies observations are i.i.d. A super learner fit is obtained for each omit-one-fold subset of the data yields predicted values for observations in the omitted fold. This procedure is invoked by setting `cvQinit = TRUE`.

The next example demonstrates the use of the `id` argument to identify observational units corresponding to subjects that contribute repeated measures. Baseline covariates are generated for 250 subjects exactly as in the previous example. These values are duplicated, and used to create a dataset of 500 observations $O = (W, A, Y, id)$, with two observations per subject.

```
R> set.seed(1960)
R> n <- 250
R> id <- rep(1:n,2)
R> W <- matrix(rnorm(n * 3), ncol = 3)
R> colnames(W) <- paste("W", 1:3, sep = "")
R> W <- rbind(W, W)
R> A <- rbinom(2 * n, 1, plogis(0.6 * W[,1] + 0.4 * W[,2] + 0.5 * W[,3]))
R> Y <- rbinom(2 * n, 1,
  plogis(A + 0.2 * W[,1] + 0.1 * W[,2] + 0.2 * W[,3]^2))
```

The data are passed to the function along with correctly specified logistic regression models as above. The only difference is that the id values generated above are supplied via the `id` argument.

```
R> result.Qcgc.repeated <- tmle(Y, A, W, family = "binomial",
+    Qform = Y ~ A + W1 + W2 + W3, gform = A ~ W1 + W2 + W3, id = id)
R> result.Qcgc.repeated

 Additive Effect
   Parameter Estimate:  0.27511
   Estimated Variance:  0.0019754
              p-value:  6.026e-10
    95% Conf Interval: (0.18799, 0.36222)

 Relative Risk
   Parameter Estimate:  1.5343
              p-value:  1.0785e-08
    95% Conf Interval: (1.3249, 1.7767)

              log(RR):  0.42805
    variance(log(RR)):  0.0056042

 Odds Ratio
   Parameter Estimate:  3.5446
              p-value:  1.0977e-08
    95% Conf Interval: (2.2966, 5.4707)

              log(OR):  1.2654
    variance(log(OR)):  0.049029
```

3.2. Stage 2: Targeting the initial estimate

The estimate of the parameter of interest can be biased when $\bar{Q}_n^0$ does not consistently estimate $\bar{Q}_0$. van der Laan and Rubin (2006) provides a theoretical foundation for constructing a parametric sub-model with fluctuation parameter ϵ that reduces residual bias that is a function of measured covariates. As mentioned above, this fluctuation involves estimating nuisance parameter g_0. Several arguments to the `tmle` function give the user control over the estimation procedure. For estimating the treatment mechanism, g_A:

- `g1W`: The conditional probability of receiving treatment given baseline covariates W.
- `gform`: A logistic regression model specification.
- `g.SL.library`: A super learner library of prediction algorithms.
- `gbound`: A value indicating symmetrical upper and lower bounds on predicted conditional treatment assignment probabilities (`gbound`, `1 - gbound`).

The first three of these are similar to the options available for estimating $\bar{Q}_0$. The `gbound` argument is a tuning parameter, conforming with the theoretical guideline that $g_n(A, W)$ must be bounded away from 0 and 1 (van der Laan and Robins 2003). Bounding will have no effect when no treatment assignments are rare within strata defined by W, e.g., `gbound` $< g_n <$ (1 $-$ `gbound`). However, when there is sparsity in the data causing a practical positivity violation, some treatment assignment probabilities will be quite small. As a consequence, some values of $H^*(A, W)$ will be very large for a subset of observations. This lack of identifiability leads to estimates with high variability. Bounding g_n away from (0,1) tends to have a beneficial effect on the variance of the resulting estimate. However, truncation introduces bias, necessitating a trade-off. These effects are most pronounced when the linear fluctuation is used for continuous outcomes, and largely mitigated by fluctuating on the logit scale (the default). Though the logistic fluctuation is strongly recommended, the package also provides a linear fluctuation option for continuous outcomes by setting the argument `fluctuation = "linear"`. Bounding g_n very close to (0,1) typically has little effect on TMLEs obtained using the logistic fluctuation. In contrast, estimates obtained using the linear fluctuation are particularly sensitive to the level of bounding of g_n.

Recall that the logistic fluctuation for continuous Y requires that Y be bounded by (a, b). When these upper and lower bounds on Y are not provided by the user via the `Qbounds` argument, the default is to use the range of the observed outcomes. This may be problematic when there is missingness in the outcome if the distribution of observed outcomes is truncated with respect to the true distribution of the outcome, thus using domain knowledge to specify bounds on $\bar{Q}_n$ is encouraged.

3.3. Examples with missing outcomes

The `Delta` argument to the `tmle` function indicates which observations have missing outcomes, with `Delta = 1` indicating that the outcome is observed. The `tmle` function ignores the Y value for observations having $\Delta = 0$, so in practice no special value is reserved to signify missing. When not explicitly specified, `Delta = 1` is assigned to all observations, signifying that no observations have missing outcomes.

When `Delta = 0` for one or more observations, the missingness mechanism is estimated from the data, or can be user-supplied. When the target parameter is $E(Y_1)$ (i.e., no treatment arms, but there is missingness in the outcome), the upper bound on g is set to 1, since $P(\Delta = 1 \mid W) = 1$, indicating no missingness within some strata of W, does not signal a positivity violation. When there are two treatment arms and some outcomes are missing, bounds on g_n apply to the product $g_n(\Delta, A, W) = g_A(A \mid W) * g_\Delta(\Delta \mid A, W)$, and the upper bound should be strictly less than 1.

The same options are available for estimating g_Δ as for estimating g_A. The relevant arguments to the `tmle` function are:

- `pDelta1`: The conditional probability of being observed given treatment assignment A and baseline covariates.
- `g.Deltaform`: Used to specify a regression formula for the regression of Δ on A and W.
- `g.SL.library`: Specifies a super learner library of prediction algorithms. The same library is used for all factors of g.

When there is no mediating variable, Z, optional argument `pDelta1`, if specified, should be an $n \times 2$ matrix, $\mathsf{P}(\Delta = 1 \mid A = 0, W), \mathsf{P}(\Delta = 1 \mid A = 1, W)$. When there is a mediating variable, an $nx4$ matrix is required: $\mathsf{P}(\Delta = 1 | Z = z, A = a, W)$, with (z, a) set to $(0,0), (0,1), (1,0), (1,1)$, respectively.

Covariates $H_0^*(A, W)$ and $H_1^*(A, W)$ for this more general data structure are given by:

$$\begin{aligned}
H_0^*(\Delta, A, W) &= \frac{I(A=0)}{g_A(0 \mid W)} \frac{1}{g_\Delta(1 \mid A, W)}, \\
H_1^*(\Delta, A, W) &= \frac{I(A=1)}{g_A(1 \mid W)} \frac{1}{g_\Delta(1 \mid A, W)},
\end{aligned}$$

and reduce to Equations 2 and 3, respectively, when there is no missingness. The fluctuation parameter ϵ is fit on observations where $\Delta = 1$. Counterfactual outcomes are obtained for all observations. Accounting for missingness increases efficiency, thus this is beneficial even when missingness is non-informative.

Population mean outcome example

The population mean outcome parameter, $\mathsf{E}(Y_1)$, is estimated when there is no variation in A for all observations, or when `A = NULL` and for some observations $\Delta = 0$. In the next example $\bar{Q}_n^0$ is based on a deliberately misspecified regression model fit on observations where $\Delta = 1$. Because a correctly specified regression model is used to estimate $\mathsf{P}(\Delta = 1 \mid W)$, bias is expected to be on the order of $1/\sqrt{n}$. At the sample size used in this example ($n = 250$), this is approximately 0.06. The true parameter value is 0.

```
R> set.seed(1960)
R> n <- 250
R> W <- matrix(rnorm(n * 3), ncol = 3)
R> colnames(W) <- paste("W",1:3, sep = "")
R> Delta <- rbinom(n, 1, plogis(0.8 + 0.3 * W[,1]))
```

```
R> Y <- 2 * W[,1] + 4 * W[,2] + 3 * W[,3] + rnorm(n)
R> Y[Delta == 0] <- NA
R> result.EY1 <- tmle(Y, A = rep(1, n), W, Qform = Y ~ W3,
+    g.Deltaform = Delta ~ W1, Delta = Delta)
R> result.EY1

 Population Mean
   Parameter Estimate:  -0.043213
   Estimated Variance:  0.15326
              p-value:  0.9121
    95% Conf Interval: (-0.81052, 0.72409)
```

3.4. Practical violations of the positivity assumption

When assignment to a particular treatment group is quite rare within some strata defined by W, the positivity assumption is technically met, however in practice this lack of information in the data (i.e., sparsity) may pose a challenging estimation problem. The next coding example illustrates typical effects of different choices of bounds on $g_n(A \mid W)$ on estimation when there is sparsity in the data. The true value for the additive treatment effect for the simulated data is $\psi_0 = 1$. Conditional treatment assignment probabilities $g_A(1 \mid W)$ range from 0.02 to 0.99. The user-supplied regression model for estimating $\bar{Q}_0$ is deliberately misspecified so that estimation is forced to rely on g. The regression formula for $g(1 \mid W)$ is correctly specified, but even so, if bounds on g_n are less than (0.05, 0.95), practical postivity violations lead to estimates with increased bias and variance when the linear fluctuation is employed, as compared to the logistic fluctuation. Parameter estimates are obtained for 250 samples of size 250.

```
R> n <- 250
R> niterations <- 250
R> gbd <- c(0, 0.01, 0.025, 0.05, 0.1)
R> ngbd <- length(gbd)
R> result.Qmgc <- matrix(NA, nrow = niterations, ncol = 2 * ngbd)
R> for(i in 1:niterations) {
+    W <- matrix(rnorm(n * 3), ncol = 3)
+    colnames(W) <- paste("W", 1:3, sep = "")
+    logitA <- 0.5 + 0.9 * W[,1] + 0.5 * W[,2] + 0.7 * W[,3]
+    A <- rbinom(n, 1, plogis(logitA))
+    Y <- A + 4 * W[,1] + 4 * W[,2] + 3 * W[,3] + rnorm(n)
+    result.Qmgc[i,] <- c(
+      unlist(sapply(gbd, function(x) {
+        tmle(Y, A, W, Qform = Y ~ A, gform = A ~ W1 + W2 + W3,
+          fluctuation = "linear", gbound = x)$estimates$ATE[1]})),
+      unlist(sapply(gbd, function(x) {
+        tmle(Y, A, W, Qform = Y ~ A, gform = A ~ W1 + W2 + W3,
+          fluctuation = "logistic", gbound = x)$estimates$ATE[1]})))
+  }
```

	Linear			Logistic		
g_n bounds	Bias	Var	MSE	Bias	Var	MSE
(0, 1)	−0.52	0.96	1.24	−0.03	0.11	0.11
(0.01, 0.99)	−0.40	0.56	0.72	−0.03	0.11	0.11
(0.025, 0.975)	−0.21	0.23	0.28	−0.03	0.09	0.09
(0.05, 0.95)	0.03	0.07	0.07	0.07	0.05	0.06
(0.1, 0.9)	0.41	0.07	0.24	0.41	0.07	0.24

Table 1: A comparison of the effect of bounding g_n using a logistic or linear fluctuation in a sparse data setting.

Results in Table 1 indicate that the bias of estimates arising from the logistic fluctuation is robust with respect to the choice of bound on g_n, until the bias introduced by bounding at (0.1, 0.9) begins to make a sizable contribution to the MSE. For this reason, respecting bounds by fluctuation the estimate on the logit scale is strongly recommended. The default bound for g is set to $(0.025, 0.975)$, but that guideline is flexible, and the effect on the bias and variance of the estimate depends on the data, e.g., if all values fall between $(0.025, 0.975)$, then setting bounds closer to $(0, 1)$ will have no effect at all.

3.5. Controlled direct effect estimation example

The first stage of the modified TMLE procedure for CDE estimates $\bar{Q}_0(Z, A, W)$. All estimation options remain available to the user: user-specified values, user-specified parametric model, super learning, cross-validated super learning. Optional user supplied values must be specified at each level of Z for each subject: the `Q` argument is used to pass in an $n \times 2$ matrix of user-determined values for $\bar{Q}_n^0(Z = 0, A, W)$. The `Q.Z1` argument is used to pass in an $n \times 2$ matrix of user-determined values for $\bar{Q}_n^0(Z = 1, A, W)$.

In the second stage $Q_n^0(Z, A, W)$ is fluctuated separately for $Z = 0$ and $Z = 1$. This requires estimation of an additional nuisance parameter, $g_\Delta = \mathsf{P}(\Delta = 1 \mid Z = z, A = a, W))$. The `pZ1` argument allows the user to pass in an $n \times 2$ matrix of conditional probabilities $\mathsf{P}(Z = 1 \mid A = 0, W), \mathsf{P}(Z = 1 \mid A = 1, W)$. Alternatively, a valid regression formula can be supplied via the `g.Zform` argument.

The following example illustrates CDE estimation in conjunction with missingness in the outcome. A sample of size 1000 is generated, with approximately 25% of outcomes missing.

```
R> n <- 1000
R> W <- matrix(rnorm(n * 3), ncol = 3)
R> colnames(W) <- paste("W", 1:3, sep = "")
R> A <- rbinom(n,1, plogis(0.6 * W[,1] + 0.4 * W[,2] + 0.5 * W[,3]))
R> Z <- rbinom(n,1, plogis(0.5 + A))
R> Y <- A + A * Z+ 0.2 * W[,1] + 0.1 * W[,2] + 0.2 * W[,3]^2 + rnorm(n)
R> Delta <- rbinom(n, 1, plogis(Z + A))
R> pDelta1 <- cbind(rep(plogis(0), n), rep(plogis(1), n),
+    rep(plogis(1), n), rep(plogis(2), n))
R> colnames(pDelta1) <- c("Z0A0", "Z0A1", "Z1A0", "Z1A1")
R> Y[Delta == 0] <- NA
```

The regression formula for estimation of $\bar{Q}_0$ is deliberately misspecified in the next call to `tmle`. Super learning is used to estimate the g_A factor of the likelihood, but the specified library contains only one algorithm, `SL.glm`, which performs a main terms regression of the outcome on all available covariates. Estimates of g_Z and g_Δ are passed in to the function. Parameter estimates are reported at each level of Z. The true parameter values are $\psi^{ATE}_{0_{Z0}} = 1$, $\psi^{ATE}_{0_{Z1}} = 2$.

```
R> result.Z.missing <- tmle(Y, A, W, Z, Delta = Delta, pDelta1 = pDelta1,
+    Qform = Y ~ 1, g.SL.library = "SL.glm")
R> result.Z.missing

Controlled Direct Effect
           ----- Z = 0 -----
 Additive Effect
   Parameter Estimate:  1.1094
   Estimated Variance:  0.034713
              p-value:  2.6122e-09
    95% Conf Interval: (0.74419, 1.4745)

           ----- Z = 1 -----
 Additive Effect
   Parameter Estimate:  1.9056
   Estimated Variance:  0.011937
              p-value:  <2e-16
    95% Conf Interval: (1.6914, 2.1197)
```

3.6. Marginal structural model example

All of the parameters discussed thus far have been estimated by calling the `tmle` function. `tmleMSM` is a second function included in the package that can be used to estimate the parameter of a user-specified MSM for binary treatment effects. This function has many arguments in common with the `tmle` function, including `Y`, `A`, `W`, `Delta`, `Q`, `Qform`, `Qbounds`, `Q.SL.library`, `cvQinit`, `gform`, `pDelta1`, `g.Deltaform`, `g.SL.library`, `family`, `fluctuation`, `alpha`, `id`, `verbose`. The user must also specify the marginal structural model via the `MSM` argument. The same flexibility for estimating each factor of the likelihood discussed above is available: user-supplied values, user-supplied regression models, and user-specified prediction algorithm libraries for data-adaptive super learning. Additional optional arguments are available:

- `V`: Covariates that can be used used to define strata within which to carry out the analysis.
- `T`: Time stamp for repeated measures data.
- `v`: Optional value defining the stratum of interest ($V = v$).
- `hAV`: Optional numerator for constructing stabilized weights.
- `hAVform`: Optional regression formula for estimating $h(A, V)$ as a regression of A on (V, T).

- `ub`: An upper bound on the weight one observation may contribute to the estimation procedure (default value is 40).

- `inference`: A flag controlling whether the variance-covariance matrix is constructed. The default value is `TRUE`, but setting `inference = FALSE` speeds up the execution time, and is recommended when bootstrapping.

The function calculates and returns the parameter estimates, the variance-covariance matrix, standard errors, pvalues, and 95% confidence intervals. The predicted values for the initial and targeted estimated counterfactual outcomes `Qinit` and `Qstar` are returned, along with estimated treatment assignment and missingness probabilities, `g`, `g_Delta`, and estimated values of $h(A, V)$, along with details of the estimation procedure. These values may be used as input to subsequent calls to the `tmle` or `tmleMSM` functions. The estimated fluctuation parameter ϵ and parameter estimates based on the untargeted initial Q are also returned, to give the user some insight into how much initial estimates differ from targeted estimates, and thus the impact of applying TMLE.

Comparison of estimators

Marginal structural models are typically fitted using inverse probability weighting (Robins 2000a; Hernan *et al.* 2000; Xiao, Abrahamowicz, and Moodie 2010). We carried out a simulation study designed to demonstrate an application of TMLE and IPTW to estimating the parameter of an MSM under two different data generating distributions. The data generation scheme is taken from a paper titled *Why Prefer Double Robust Estimators?* (Neugebauer and van der Laan 2005), that discusses the use of inverse probability of treatment weighting (IPTW) and double robust augmented IPTW (AIPTW) estimators. One of the instructive lessons from that paper is that under near-positivity violations leading to large inverse probability weights, a double robust estimator can out-perform IPTW even when the propensity score model and the MSM are correctly specified. Since that paper pre-dates the introduction of TMLE, TMLE was not included in the comparison.

The observed data structure is given by $O = (W, A, Y)$. W and Y are continuous random variables that are functions of an unobserved covariate, U, that does not confound the effect of treatment on the outcome. Treatment assignment is a function of W. We are interested in estimating the two-dimensional parameter β of an MSM given by $Y = \beta_0 + \beta_1 A$, where the true value of $\beta = (\beta_0, \beta_1) = (2, -5)$. Three estimators were applied to this problem, the `tmleMSM` function, the IPTW estimator using unstabilized weights, $wt_i = 1/g_n(A_i \mid W_i$ and a second IPTW estimator using stabilized weights, $wt_{i,stab} = \left[\frac{1}{n}\sum_{i=1}^{n}(A = A_i)\right]/g_n(A_i \mid W_i)$.

Two treatment assignment mechanisms were defined that differ in the strength of the association between A and W.

$$\begin{aligned} g_1 &= P(A = 1 \mid W) = \text{expit}(0.1 + 0.25W) \\ g_2 &= P(A = 1 \mid W) = \text{expit}(1 + 1.5W) \end{aligned}$$

The empirical probability of receiving treatment according to mechanism g_1 ranges between approximately 0.16 and 0.88, corresponding to inverse weights of 1.14 and 6.25. These values indicate that except possibly at extremely small sample size, no observation would receive enough weight to completely dominate the analysis. In contrast, treatment assignment probabilities based on mechanism g_2 range between 6×10^{-5} and 0.999995. We call this a near-positivity violation, and it poses a challenging estimation scenario.

Notice that under this specification of the MSM, the coefficient β_1 is equivalent to the ATE parameter estimated by the `tmle` function. We would expect the estimate of β_1 obtained using the `tmleMSM` function to be equal to the estimate of the ATE parameter returned by the `tmle` function (allowing for a certain imprecision due to the differences in the way the calculations are carried out). The simulation results bear this out. We also estimate the ATE parameter using an alternative double-robust estimator, the augmented IPTW estimator (AIPTW) introduced in Robins and Rotnitzky (1992). The AIPTW estimator is defined as,

$$\psi_n^{AIPTW} = \frac{1}{n}\sum_{i=1}^{n}\frac{[I(A_i=1)-I(A_i=0)]}{g_n(A_i \mid W_i)}(Y_i - \bar{Q}_n^0(A_i,W_i)) + \frac{1}{n}\sum_{i=1}^{n}(\bar{Q}_n^0(1,W_i) - \bar{Q}_n^0(0,W_i)).$$

R code that defines a function to calculate AIPTW estimates of the ATE parameter given dataset `d`, outcome regression model `Qform`, and estimated treatment assignment probabilities, `g1W` is shown next.

```
R> calc_aipw <- function(d, Qform, g1W) {
+    Q <- glm(Qform, data = d)
+    QAW.pred <- predict(Q)
+    Q1W.pred <- predict(Q, newdata = data.frame(d[,-2], A = 1))
+    Q0W.pred <- predict(Q, newdata = data.frame(d[,-2], A = 0))
+    h <- d$A/g1W - (1- d$A)/(1 - g1W)
+    return(psi = mean(h*(d$Y - QAW.pred) + Q1W.pred - Q0W.pred))
+  }
```

The next code chunk runs the Monte Carlo simulation study. The outside loop corresponds to the two treatment assignment mechanisms. Within the inner loop a dataset is generated at each iteration and subsequently analyzed. Models for the MSM and the conditional distribution of the treatment assignment are correctly specified, and the same (unbounded) predicted treatment assignment probabilities are used by each estimator. The specification of the model for Q is slightly misspecified for TMLE and AIPTW estimators, by omitting the unobserved covariate, U, but this omission does not bias the estimate of the ATE parameter.

```
R> set.seed(10)
R> n <- 500
R> niter <- 500
R> a <- c(.1, 1)
R> b <- c(.25,1.5)
R> est.beta0 <- array(NA, dim = c(2, niter, 3),
+    dimnames = list(c("g1",  "g2"), NULL,
+    c("IPW", "IPW stabilized", "TMLE.MSM")))
R> est.ATE <- array(NA, dim = c(2, niter, 5),
+    dimnames = list(c("g1", "g2"), NULL,
+    c("IPW", "IPW stabilized", "TMLE.MSM", "TMLE", "AIPW")))
R> for (i in 1:2) {
+    for (j in 1:niter) {
```

```
+       U <- runif(n, -10, 10)
+       W <- U/3 + rnorm(n)
+       logitA <- a[i] + b[i]*W
+       A <- rbinom(n, 1, plogis(logitA))
+       Y <- 2 + 4 * U - 5 * A + rnorm(n)
+       g <- glm(A ~ W, family = "binomial")
+       g1W <- predict(g, type = "response")
+       wt <- A/g1W + (1 - A)/(1 - g1W)
+       wt.stab <- (A * mean(A) + (1 - A) * (1 - mean(A))) * wt
+       ipw.msm <- coef(glm(Y ~ A, weights = wt))
+       ipw.stab.msm <- coef(glm(Y ~ A, weights = wt.stab))
+       res.tmleMSM <- tmleMSM(Y, A, as.matrix(W), V = rep(1, n), MSM = "A",
+         Qform = "Y ~ A", g1W = g1W, ub = Inf)
+       res.tmle <- tmle(Y, A, as.matrix(W), Qform ="Y ~ A", g1W = g1W,
+         gbound = c(0,1))
+       aipw <- calc_aipw(data.frame(Y, A, W), Qform = "Y ~ A", g1W = g1W)
+       est.beta0[i,j,] <- c(ipw.msm[1], ipw.stab.msm[1], res.tmleMSM$psi[1])
+       est.ATE[i,j,] <- c(ipw.msm[2], ipw.stab.msm[2], res.tmleMSM$psi[2],
+         res.tmle$estimates$ATE$psi, aipw)
+   }}
```

Results displayed in Table 2 and Figure 1 indicate that all estimators perform well under treatment assignment mechanism g_1. When the more extreme treatment assignment mechanism is used to generate the data (g_2), performance of the IPTW estimators and AIPTW degrades significantly, while both TMLEs exhibit more moderate increases in bias and variance.

TMLE is a substitution estimator that ensures global bounds of the statistical model are respected, thereby constraining the bias and variance. Both AIPTW and TMLE solve the same estimating equation and are asymptotically equivalent estimators of the ATE parameter. However, as illustrated by the plots in the figure and the results reported in Table 2, depending on the characteristics of the underlying data distribution the difference in their finite sample

	g_1 *(no pos. violation)*			g_2 *(near pos. violation)*		
	Bias	Var	MSE	Bias	Var	MSE
$\beta_0 = 2$						
IPW	-0.006	1.31	1.31	-2.25	48.40	53.39
IPW stabilized	-0.006	1.31	1.31	-2.25	48.40	53.39
TMLE.MSM	-0.006	1.28	1.28	0.16	11.29	11.30
$\beta_1 = -5$						
IPW	0.005	0.93	0.93	3.00	48.28	57.20
IPW stabilized	0.005	0.93	0.93	3.00	48.28	57.20
TMLE.MSM	0.006	0.93	0.93	-0.18	12.00	12.01
$ATE = -5$						
TMLE	0.006	0.93	0.93	-0.18	12.00	12.01
AIPW	0.004	0.93	0.93	0.56	322.33	322.00

Table 2: Estimator comparison, $n = 500$.

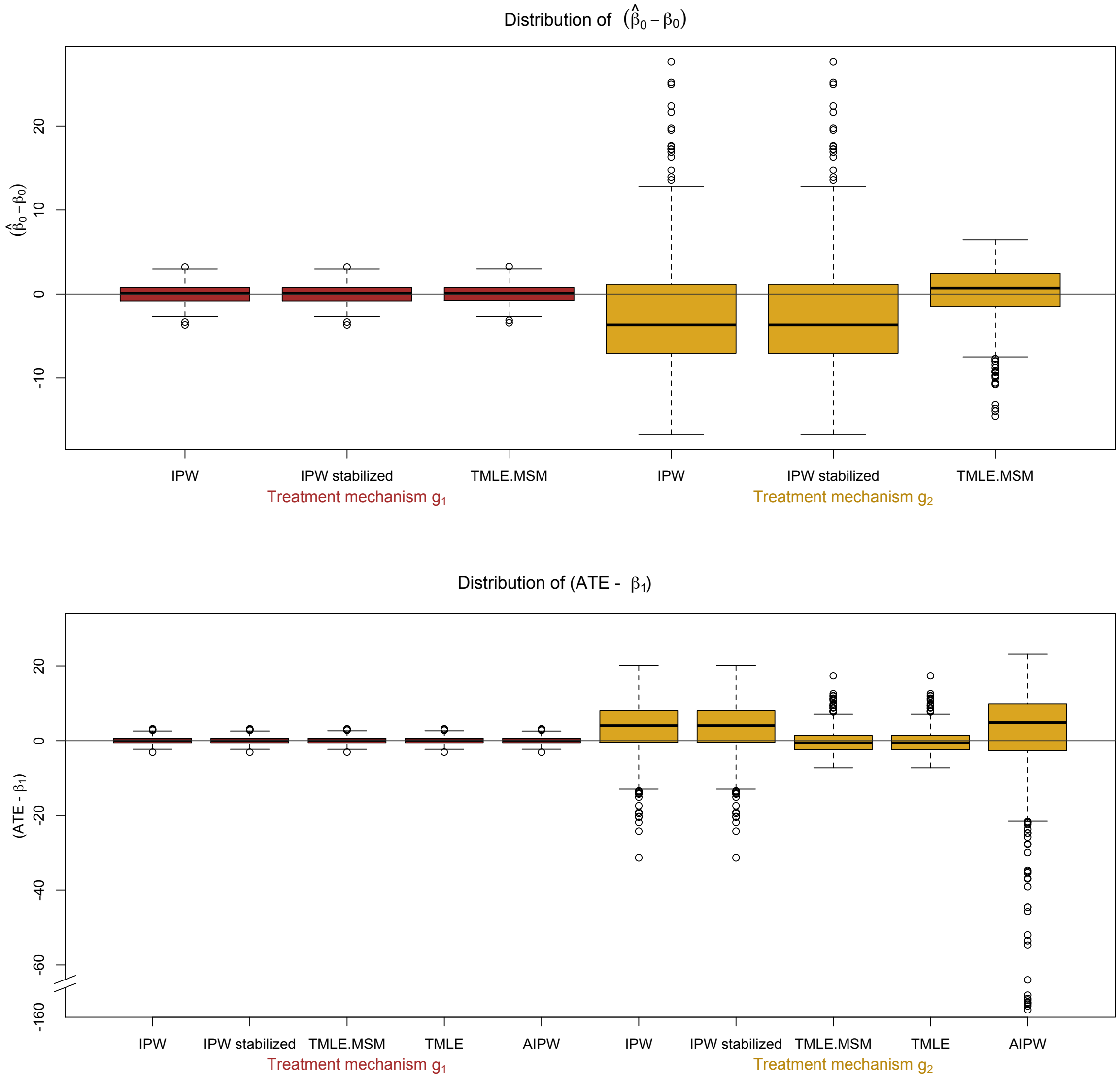

Figure 1: Distribution of parameter estimates minus the true parameter value under two different treatment assignment mechanisms, no positivity violations (g_1, left), and practical positivity violations (g_2, right).

performance can be striking. An in-depth discussion of the relative performance of TMLE in comparison with other double robust estimators discussed in the literature can be found in Porter *et al.* (2011).

4. FEV data analysis

TMLE was applied to assess the marginal effect of smoking on forced expiratory volume (FEV) using data originally introduced in Rosner (1999b) and discussed in Kahn (2005). The data consists of 654 observations with five variables recorded for each subject: `age` (years), fev (liters), ht (height in inches), sex (0 = female, 1 = male), smoke (0 = non smoker, 1 = smoker) (Rosner 1999a). FEV is a measure of pulmonary function that is related to body size and lung capacity. Thus, the relationship between smoking and FEV is likely to be confounded by age and sex, both of which influence FEV and are associated with smoking status. Though height does not have an obvious link to smoking behavior accounting for covariates predictive of the outcome can improve efficiency, so we include it in the analysis. The data are from an observational study of children 3–19 years old. No children younger than nine years old smoked cigarettes. Therefore, any attempt to estimate a marginal effect of smoking on FEV adjusted for age incurs a theoretical positivity violation due to a complete lack of support in the data. For this reason we restrict the analysis to the subset of data containing $n = 439$ observations on subjects ages 9–19.

The observed data consists of n i.i.d. copies of $O = (W, A, Y) \sim \mathsf{P}_0$, where $W =$ (age, ht, sex), A is an indicator of smoking status, and Y is a continuous measure of FEV. The outcome of interest is the marginal additive effect of smoking on FEV, defined as $E_W[E(Y \mid A = 1, W) - E(Y \mid A = 0, W)]$. If the true regression of Y on A and W were a main terms linear regression, this parameter would correspond to the coefficient in front of the treatment term. However, there is no reason to believe that is the case, and an estimate of the treatment effect based on this misspecified model for $\bar{Q}$ is likely to be biased. The double-robustness property of TMLE tells us that even given a misspecified $\bar{Q}_n^0$, the targeting step can reduce this bias, given a consistent estimate of the treatment mechanism. In the next example we deliberately supply a main terms model for $\bar{Q}$ that we assume is misspecified, and use super learning to estimate $g_A(1 \mid W)$. The algorithms included in the super learner library are:

- SL.glm: Main terms logistic regression of A on W (R Development Core Team 2012).
- SL.step: Stepwise forward and backward model selection using AIC criterion, restricted to second order polynomials (R Development Core Team 2012).
- SL.DSA.2: DSA algorithm searching over second order polynomials, substitution and addition moves enabled (Neugebauer and Bullard 2010).
- SL.loess: Local fitting of a polynomial response surface (span = 0.75) (R Development Core Team 2012).
- SL.caret: Random forest, with data-adaptively selected value for mtry parameter (Kuhn 2008).
- SL.bart: A classifier based on a Bayesian sum-of-trees model with ntree = 300 (Chipman and McCulloch 2010).
- SL.knn, SL.knn20, SL.knn40, SL.knn60: k-nearest neighbor algorithm, with neighborhood size, k, set to 10, 20, 40, 60 (Venables and Ripley 2002).

```
R> data("fev")
R> fev <- fev[fev$age >= 9, ]
R> g.SL.library <- c("SL.glm", "SL.step", "SL.DSA.2","SL.loess", "SL.caret",
+    "SL.bart", "SL.knn", "SL.knn20", "SL.knn40", "SL.knn60")
R> smoke.Qmis <- tmle(Y = fev$fev, A = fev$smoke, W = fev[, c(1, 3, 4)],
+    Qform = Y ~ ., g.SL.library = g.SL.library)
R> smoke.Qmis
```

```
 Additive Effect
   Parameter Estimate:  -0.099653
   Estimated Variance:  0.0045071
              p-value:  0.13771
    95% Conf Interval: (-0.23124, 0.031932)
```

The parameter estimate after targeting is $1/n \sum_{i=1}^{n} \bar{Q}_n^*(1, W_i) - \bar{Q}_n^*(0, W_i) = -0.10$. Users are often curious about how targeting affects the parameter estimate. The function returns initial (untargeted) predicted values, $\bar{Q}_n^0(0, W), \bar{Q}_n^0(1, W)$. This allows the user to calculate a parameter estimate of -0.16 based on the initial estimate of $\bar{Q}_0$ as follows:

```
R> EY0 <- mean(smoke.Qmis$Qinit$Q[,"Q0W"])
R> EY1 <- mean(smoke.Qmis$Qinit$Q[,"Q1W"])
R> EY1 - EY0

[1] -0.1574331
```

Recall that TMLE is asymptotically efficient when both $\bar{Q}_0$ and g_0 are estimated consistently. In the next example, super learning is used to estimate $\bar{Q}_0$ data-adaptviely. The prediction algorithm library includes all the algorithms specified for the estimation of $g$ that do not require a binary outcome (everything except the k-nearest neighbor algorithms), and also a linear regression of Y on A and W that includes main terms and all interactions of A and W. We begin by defining a new super learner wrapper function, `SL.glm.int`:

```
R> SL.glm.int <- function(Y.temp, X.temp, newX.temp, family, ...) {
+    Aint <- paste("A",  colnames(X.temp)[-c(1, 2)], sep = "*")
+    form <- paste("Y.temp ~ Z + ", paste(Aint, collapse = "+"))
+    fit.glm <- glm(form, data = data.frame(Y.temp, X.temp), family = family)
+    out <- predict(fit.glm, newdata = newX.temp, type = "response")
+    fit <- list(object = fit.glm)
+    foo <- list(out = out, fit = fit)
+    class(foo$fit) <- c("SL.glm.int")
+    return(foo)
+  }
R> Q.SL.library <- c("SL.glm", "SL.glm.int", "SL.DSA.2", "SL.loess",
+    "SL.caret", "SL.bart")
```

The library for estimating $\bar{Q}_0$ is passed into the `tmle` function. Because the predicted values for $g_A(1 \mid W)$ are not affected by altering the method used to estimate $\bar{Q}_0$, this next example illustrates a way to reduce computation time by passing in the treatment assignment probabilities obtained from the previous invocation of the function.

```
R> smoke.QSL <- tmle(Y = fev$fev, A = fev$smoke, W = fev[, c(1, 3, 4)],
+    Q.SL.library = Q.SL.library, g1W = smoke.Qmis$g$g1W)
R> smoke.QSL

Additive Effect
   Parameter Estimate:  -0.082194
```

```
 Estimated Variance:  0.0037794
            p-value:  0.18122
  95% Conf Interval: (-0.20269, 0.0383)
```

When a data-adaptive approach to estimating $\bar{Q}_0$ is used, the parameter estimate of -0.08 is quite close to -0.10, the estimate obtained when TMLE was forced to incorporate the (presumably) misspecified model for $\bar{Q}_n^0$. Super learning also improved efficiency.

Stage one of the TMLE procedure is concerned with explaining variance in the outcome. Because ψ_0 is a function of the Q portion of the likelihood, improving the estimate of $\bar{Q}_0$ tends to improve the estimate of ψ_0. However, estimation procedures for $\bar{Q}_0$ have a different goal with respect to the bias/variance tradeoff than do estimators of ψ_0. TMLE's goal is to optimize the tradeoff with respect to ψ_0.

5. Discussion

The **tmle** package was designed to provide a flexible, easily customizable implementation of TMLE for binary point treatment effects. A novice user has only to supply the data, while advanced users can control the estimation procedure by overriding default specifications and/or supplying values for $\bar{Q}_n^0$ and g_n from any external estimation procedure. The function can internally estimate any factor of the likelihood with user-supplied linear or logistic regression models, or can use super learning to obtain data-adaptive fits. Covariate information is exploited to reduce bias and increase efficiency in estimates when outcome data is missing. Influence curve-based inference readily accounts for repeated measures. The ability to incorporate data-adaptive machine learning techniques while still providing valid inference is an additional desirable feature of TMLE.

Planned extensions to the package include incorporating external weights on observations, estimating additional parameters, such as the average treatment effect among the treated (ATT). Additional loss functions and fluctuation models that increase robustness with respect to outliers and sparsity are under development. TMLE applications to estimating causal effects of multiple time-point interventions while controlling for time-dependent covariates are also under development. Beta versions of the code are often made available at `http://www.stat.berkeley.edu/~laan/Software/` before they are incorporated into the **tmle** package. Another open area of research is finding an optimal strategy for nuisance parameter estimation. van der Laan and Gruber (2010) presents a theorem on collaborative double robustness of the efficient influence curve that sheds light on this problem. The theorem indicates that depending on the difference $(Q_n - Q_0)$, in addition to g_0 there may exist one or more conditional nuisance parameter distributions that together with the initial estimate solve the estimating equation at the true parameter value, ψ_0. The paper describes a collaborative targeted forward selection algorithm for fitting g that is guided by the goodness-of-fit for the corresponding TMLE of Q_0, and thus on its utility for estimating ψ_0 (see also Gruber and van der Laan 2010a). A beta version of R software for collaborative TMLE (C-TMLE) is available (Gruber 2010). TMLE has been successfully applied to the analysis of time-to-event data (Moore and van der Laan 2009; Stitelman and van der Laan 2010), data from sequentially randomized trials (Chambaz 2011), and other application areas, however this capability is not yet available in the **tmle** package.

6. Answers to some frequently asked questions (FAQ)

Can the treatment variable be categorical or continuous?

The package only handles treatment encoded by a binary treatment indicator, A. However, if treatment is categorical or can be discretized, causal contrasts can still be evaluated by viewing the analysis as a missing data problem. From this perspective we can estimate the marginal mean outcome $E(Y_a)$ when treatment is set to a particular level, a for the entire population. By the consistency assumption, the outcome recorded in the dataset for observations where $A = a$ is understood to be observed, while the outcome for observations where $a \neq a$ is defined as missing. TMLE is applied to estimate the $E(Y_a)$ parameter in this dataset, with a taking on each value in $\mathcal{A}$, the set of all treatments, in turn. This procedure yields estimated marginal mean outcomes under each treatment level. With these in hand, the analyst can compute two-way causal contrasts and estimate a dose-response relationship. The next coding example illustrates how this is done. First observations $O = (W, A, Y)$ are generated such that A takes on the value 1, 2, or 3 with uniform probability. Next binary missingness indicator variables are defined such that $\Delta_a = 1$ if$A = a$ and 0 otherwise.

```
R> set.seed(10)
R> n <- 10^5
R> A <- sample(1:3, n, replace = TRUE)
R> W1 <- rnorm(n)
R> W2 <- rbinom(n, 1, 0.3)
R> Y <- 2 * A + 10 * W1 - A * W2 + rnorm(n)
R> Delta1 <- as.integer(A == 1)
R> Delta2 <- as.integer(A == 2)
R> Delta3 <- as.integer(A == 3)
```

Next we obtain three estimates of the marginal mean outcome had the whole population been assigned to treatment at a particular level. We deliberately misspecify the regression formula for $\bar{Q}_n^0$ in each case, while correctly specifying the missingness mechanism, $P(\Delta_a = 1|W)$. The true marginal treatment effects are (1.7, 3.4, 5.1) at levels $a = (1, 2, 3)$, respectively.

```
R> result1 <- tmle(Y, NULL, cbind(W1, W2), Delta = Delta1,
+    Qform = "Y ~ A", g.Deltaform = "Delta ~ 1")
R> result2 <- tmle(Y, NULL, cbind(W1, W2), Delta = Delta2,
+    Qform = "Y ~ A", g.Deltaform = "Delta ~ 1")
R> result3 <- tmle(Y, NULL, cbind(W1, W2), Delta = Delta3,
+    Qform = "Y ~ A", g.Deltaform = "Delta ~ 1")
R> print(c(result1$estimates$EY1$psi, result2$estimates$EY1$psi,
+    result3$estimates$EY1$psi))
```

```
[1] 1.636907 3.402428 5.112978
```

Any causal contrast of these three parameters, EY_1, EY_2, EY_3 can be computed. Calculation of the additive effect corresponding to receiving treatment at level 1 vs. level 3 is shown next. We also show how a dose response measure can be estimated by regressing the targeted predicted outcomes on the corresponding treatment assignment. The true linear dose-response is 1.7.

```
R> print(result1$estimates$EY1$psi - result3$estimates$EY1$psi)

[1] -3.476071

R> Qstar <- c(result1$Qstar[,1], result2$Qstar[,1], result3$Qstar[,1])
R> A.intervened <- rep(1:3, each = n)
R> doseResponse <- coef(lm(Qstar ~ A.intervened))[2]
R> print(doseResponse)

A.intervened
    1.738036
```

Observations where the predicted outcomes arise from setting A to a level corresponding to no treatment should be omitted from the data used to fit the dose-response regression.

Is there a way to see the parameter estimates based on the initial (untargeted) estimate $\bar{Q}_n^0$?

The `tmle` function returns the initial estimates for $\bar{Q}(0, W), \bar{Q}(1, W)$, as a matrix, `result$Qinit$Q`. $\mathsf{E}(Y_0)$ can be estimated as `mean(Qinit$Q[, "Q0W"])`, $\mathsf{E}(Y_1)$ can be estimated as `mean(Qinit$Q[, "Q1W"])`, From there any desired parameter estimate can be calculated. For CDE estimation, `result[[1]]$Qinit$Q` corresponds to values obtained when $Z = 0$, and `result[[2]]$Qinit$Q` corresponds to values obtained by setting $Z = 1$.

Can I use the package for count data (poisson regression)?

Data-adaptive estimation of $\bar{Q}_0$ is not available for count data, but the package can estimate the additive effect of point treatment on a poisson-distributed outcome variable by supplying a formula for poisson regression (log link only), and setting `family = "poisson"`. The fluctuation will be carried out on the logit scale, unless `fluctuation = "linear"` is specified. In this case, despite the name, poisson regression will be used to fit ϵ. If data-adaptive estimation of $\bar{Q}_0$ is desired, specify `family = "gaussian"`, and externally enforce the constraint that predicted values cannot be less than 0 by specifying `Qbounds = c(0, ub)`, with an appropriate value filled in for the upper bound. Although this will ensure that the initial estimate of the conditional mean outcome is non-negative, unless the logistic fluctuation is used there is no guarantee that the targeted estimate will respect this constraint.

Can I call the `tmle` *function a second time without having to re-do the initial estimation of $\bar{Q}_0$?*

Yes. Predicted values based on the initial estimate $\bar{Q}_0$ are returned as `result$Qinit$Q` (assuming the result of the first call to `tmle` was assigned to the variable named `result`). These values can be passed into a second call to `tmle` by specifying a value for the `Q` argument: `Q = result$Qinit$Q`. For CDE estimation, values for two arguments must be supplied, `Q = result[[1]]$Qinit$Q, Q.Z1 = result[[2]]$Qinit$Q`.

Values for the conditional probabilities for treatment assignment, intermediate variable, and missingness are also available to be examined or passed into a second invocation of `tmle`: `g1W = result$g$g1W, pZ1 = result$g.Z$g1W, pDelta1 = result$g.Delta$g1W`. These are untruncated values, regardless of the value of the `gbound` argument.

*Can the **tmle** package handle time-to-event data, (e.g., Cox or AFT models)?*

TMLE has been applied to the analysis of time-to-event data (Moore and van der Laan 2009; Stitelman and van der Laan 2010), however the package currently does not offer survival analysis.

How does TMLE compare to other methods in the causal inference literature?

Comparisons of TMLE performance with other estimators, including inverse probability of treatment weighting (IPTW), propensity score based methods, and other double robust estimators can be found in papers in many statistical journals. A book on targeted minimum loss-based learning is available (van der Laan and Rose 2011), a collection of papers on TMLE through 2009 can be downloaded from `http://www.bepress.com/ucbbiostat/sgruber/6` (van der Laan *et al.* 2009). Additional references are listed in Section 1.1.

Acknowledgments

This work was supported by NIH grant R01 A1074345-04 and the National Institutes of Health/National Heart, Lung and Bood Institute grant R01 HL080644.

References

Bang H, Robins JM (2005). "Doubly Robust Estimation in Missing Data and Causal Inference Models." *Biometrics*, **61**, 962–72.

Bickel PJ, Klaassen CAJ, Ritov Y, Wellner J (1997). *Efficient and Adaptive Estimation for Semiparametric Models*. Springer-Verlag.

Chambaz A (2011). "TMLE in Adaptive Group Sequential Covariate-Adjusted RCTs." In van der Laan and Rose (2011), chapter 29.

Chipman H, McCulloch R (2010). ***BayesTree**: Bayesian Methods for Tree Based Models*. R package version 0.3-1.1, URL `http://CRAN.R-project.org/package=BayesTree`.

Gill RD, van der Laan MJ, Robins JM (1997). "Coarsening At Random: Characterizations, Conjectures and Counter-Examples." In DY Lin, TR Fleming (eds.), *Proceedings of the First Seattle Symposium in Biostatistics*, pp. 255–94. Springer-Verlag, New York.

Gruber S (2010). *Collaborative Targeted Maximum Likelihood Estimation*. R Software Version 0.5, URL `http://www.stat.berkeley.edu/~laan/Software/`.

Gruber S, van der Laan MJ (2010a). "An Application of Collaborative Targeted Maximum Likelihood Estimation in Causal Inference and Genomics." *The International Journal of Biostatistics*, **6**(1).

Gruber S, van der Laan MJ (2010b). "A Targeted Maximum Likelihood Estimator of a Causal Effect on a Bounded Continuous Outcome." *The International Journal of Biostatistics*, **6**(1).

Hampel FR (1974). "The Influence Curve and Its Role in Robust Estimation." *Journal of the American Statistical Association*, **69**(346), 383–93.

Heitjan DF, Rubin DB (1991). "Ignorability and Coarse Data." *The Annals of Statistics*, **19**(4), 2244–2253.

Hernan MA, Brumback B, Robins JM (2000). "Marginal Structural Models to Estimate the Causal Effect of Zidovudine on the Survival of HIV-Positive Men." *Epidemiology*, **11**(5), 561–570.

Jacobsen M, Keiding N (1995). "Coarsening at Random in General Sample Spaces and Random Censoring in Continuous Time." *The Annals of Statistics*, **23**, 774–86.

Kahn M (2005). "An Exhalent Problem for Teaching Statistics." *The Journal of Statistical Education*, **13**(2).

Kuhn M (2008). "Building Predictive Models in R Using the **caret** Package." *Journal of Statistical Software*, **28**(5), 1–26. URL http://www.jstatsoft.org/v28/i05/.

Moore KL, van der Laan MJ (2009). "Application of Time-to-Event Methods in the Assessment of Safety in Clinical Trials." In KE Peace (ed.), *Design, Summarization, Analysis & Interpretation of Clinical Trials with Time-to-Event Endpoints*. Chapman & Hall/CRC, Boca Raton.

Neugebauer R, Bullard J (2010). ***DSA**: Deletion/Substitution/Addition Algorithm*. R package version 3.1.4, URL http://www.stat.berkeley.edu/~laan/Software/.

Neugebauer R, van der Laan MJ (2005). "Why Prefer Double Robust Estimators in Causal Inference?" *Journal of Statistical Planning and Inference*, **129**(1–2), 405–426.

Pearl J (2010a). "The Causal Foundations of Structural Equation Modeling." *Technical Report R-370*, University of California, Los Angeles, Department of Computer Science.

Pearl J (2010b). "An Introduction to Causal Inference." *The International Journal of Biostatistics*, **6**(2).

Polley EC, van der Laan MJ (2012). ***SuperLearner**: Super Learner Prediction*. R package version 2.0-9, URL http://CRAN.R-project.org/package=SuperLearner.

Porter KE, Gruber S, van der Laan MJ, Sekhon JS (2011). "The Relative Performance of Targeted Maximum Likelihood Estimators." *The International Journal of Biostatistics*, **7**(31), 1–34.

R Development Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Robins JM (1986). "A New Approach to Causal Inference in Mortality Studies with Sustained Exposure Periods – Application to Control of the Healthy Worker Survivor Effect." *Mathematical Modelling*, **7**, 1393–1512.

Robins JM (1997). "Marginal Structural Models." In *1997 Proceedings of the American Statistical Association Section on Bayesian Statistical Science*, pp. 1–10.

Robins JM (2000a). "Marginal Structural Models versus Structural Nested Models as Tools for Causal Inference." In *Statistical Models in Epidemiology, the Environment, and Clinical Trials (Minneapolis, MN, 1997)*, pp. 95–133. Springer-Verlag, New York.

Robins JM (2000b). "Robust Estimation in Sequentially Ignorable Missing Data and Causal Inference Models." In *Proceedings of the American Statistical Association: Section on Bayesian Statistical Science*, pp. 6–10.

Robins JM, Hernan MA, Brumback B (2000a). "Marginal Structural Models and Causal Inference in Epidemiology." *Epidemiology*, **11**(5), 550–560.

Robins JM, Rotnitzky A (1992). "Recovery of Information and Adjustment for Dependent Censoring Using Surrogate Markers." In NP Jewell, K Dietz, B Farewell (eds.), *AIDS Epidemiology*. Bikhäuser, Boston.

Robins JM, Rotnitzky A (2001). "Comment on the Bickel and Kwon Article, 'Inference for Semiparametric Models: Some Questions and an Answer'." *Statistica Sinica*, **11**(4), 920–936.

Robins JM, Rotnitzky A, van der Laan MJ (2000b). "Comment on 'On Profile Likelihood'." *Journal of the American Statistical Association*, **450**, 431–435.

Rosenblum M (2011). "Marginal Structural Models." In van der Laan and Rose (2011), chapter 9.

Rosenblum M, van der Laan MJ (2010). "Targeted Maximum Likelihood Estimation of the Parameter of a Marginal Structural Model." *The International Journal of Biostatistics*, **6**(19).

Rosner B (1999a). "FEV Dataset." Submitted by MJ Kahn, Wheaton College, Norton, MA, URL `http://www.amstat.org/publications/jse/v13n2/datasets.kahn.html`.

Rosner B (1999b). *Fundamentals of Biostatistics*. 5th edition. Duxbury Press, Pacific Grove.

Rubin DB (1974). "Estimating Causal Effects of Treatments in Randomized and Nonrandomized Studies." *Journal of Educational Psychology*, **64**, 688–701.

Scharfstein DO, Rotnitzky A, Robins JM (1999). "Adjusting for Non-Ignorable Drop-Out Using Semiparametric Nonresponse Models." *Journal of the American Statistical Association*, **94**(448), 1096–1120.

Sinisi S, van der Laan MJ (2004). "The Deletion/Substitution/Addition Algorithm in Loss Function Based Estimation: Applications in Genomics." *Journal of Statistical Methods in Molecular Biology*, **3**(1).

Stitelman OM, van der Laan MJ (2010). "Collaborative Targeted Maximum Likelihood for Time to Event Data." *The International Journal of Biostatistics*, **6**(1).

van der Laan MJ, Gruber S (2010). "Collaborative Double Robust Penalized Targeted Maximum Likelihood Estimation." *The International Journal of Biostatistics*, **6**(1).

van der Laan MJ, Gruber S (2011). "Targeted Maximum Loss Based Estimation of an Intervention Specific Mean." *Technical Report 290*, Division of Biostatistics, University of California, Berkeley. URL http://www.bepress.com/ucbbiostat/paper290.

van der Laan MJ, Polley E, Hubbard A (2007). "Super Learner." *Statistical Applications in Genetics and Molecular Biology*, **6**(25).

van der Laan MJ, Robins JM (2003). *Unified Methods for Censored Longitudinal Data and Causality*. Springer-Verlag.

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

van der Laan MJ, Rose S, Gruber S (2009). "Readings in Targeted Maximum Likelihood Estimation." *Technical Report 254*, Division of Biostatistics, University of California, Berkeley. URL http://www.bepress.com/ucbbiostat/paper254.

van der Laan MJ, Rubin D (2006). "Targeted Maximum Likelihood Learning." *The International Journal of Biostatistics*, **2**(1).

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition. Springer-Verlag, New York.

Xiao Y, Abrahamowicz M, Moodie EEM (2010). "Accuracy of Conventional and Marginal Structural Cox Model Estimators: A Simulation Study." *The International Journal of Biostatistics*, **6**.

A. Influence curve-based inference

Theory tells us that the difference between a parameter estimate obtained from an RAL estimator and the true parameter value converges at a root-n rate to a Normal limit distribution, $\sqrt{n}(\psi_n - \psi_0) \overset{D}{\to} N(0, \Sigma)$, where Σ is the covariance matrix of the (possibly multi-dimensional) parameter (Bickel *et al.* 1997). In practice, this provides a means for estimating the variance of the estimator as the variance of the empirical influence curve divided by the number of i.i.d. units of observation, n. Parameter-specific influence curves that the software uses as the basis for calculating p values and 95% confidence intervals are given below. Asymmetric confidence intervals for the RR and OR parameters are constructed on the log scale, based on the influence curves for the log(RR) and log(OR), respectively.

$$\begin{aligned} IC^{EY_1}(O) &= \frac{\Delta}{g_{0_\Delta}(1 \mid W)}(Y - \bar{Q}_0(W)) + \bar{Q}_0(W) - \psi_0^{EY_1} \\ IC^{ATE}(O) &= \left(\frac{A}{g_{0_A}(1 \mid W)} - \frac{1-A}{g_{0_A}(0 \mid W)}\right)\frac{\Delta}{g_{0_\Delta}(1 \mid A, W)}(Y - \bar{Q}_0(A,W)) \\ &\quad + \bar{Q}_0(1,W) - \bar{Q}_0(A,W) - \psi_0^{ATE} \end{aligned}$$

$$\begin{aligned} IC^{logRR}(O) &= \frac{1}{\mu_1}\left(\frac{A}{g_{0_A}(1 \mid W)}\frac{\Delta}{g_{0_\Delta}(1 \mid A, W)}(Y - \bar{Q}_0(A,W)) + \bar{Q}_0(1,W) - \mu_1\right) \\ &\quad -\frac{1}{\mu_0}\left(\frac{1-A}{1 - g_{0_A}(1 \mid W)}\frac{\Delta}{g_{0_\Delta}(1 \mid A, W)}(Y - \bar{Q}_0(A,W)) + \bar{Q}_0(0,W) - \mu_0\right) \end{aligned}$$

$$\begin{aligned} IC^{logOR}(O) &= \frac{1}{\mu_1(1-\mu_1)}\left(\frac{A}{g_{0_A}(1 \mid W)}\frac{\Delta}{g_{0_\Delta}(1 \mid A, W)}(Y - \bar{Q}_0(A,W)) + \bar{Q}_0(1,W)\right) \\ &\quad -\frac{1}{\mu_0(1-\mu_0)}\left(\frac{1-A}{1 - g_{0_A}(1 \mid W)}\frac{\Delta}{g_{0_\Delta}(1 \mid A, W)}(Y - \bar{Q}_0(A,W)) + \bar{Q}_0(0,W)\right) \end{aligned}$$

Each IC is evaluated by substituting estimates of the true unknown quantities in the above formulas, $\hat{\mu}_0, \hat{\mu}_1, g_{n_A}, g_{n_\Delta}$, and in particular, the targeted estimate $\bar{Q}_n^*(A,W)$ in place of $\bar{Q}_0(A,W)$. A conservative estimate of the variance of the parameter estimate is given by $\hat{\sigma}^2 = \mathsf{VAR}(\widehat{IC}(O))/n$, where n is the number of i.i.d. units of observation. If the dataset contains repeated measures on independent subjects, the subject is considered the unit of observation, and the unit's contribution to the influence curve is equal to the mean contribution for that subject. Ninety-five percent confidence intervals are calculated as $\psi_n(Q_n^*) \pm 1.96\hat{\sigma}/\sqrt{n}$ for the ATE and EY$_1$ parameters, and $\exp(\log(\psi_n(Q_n^*)) \pm 1.96\hat{\sigma}/\sqrt{n})$ for the RR and OR parameters, with $\hat{\sigma}$ equal to the estimated standard error of the $\log(RR)$ or $\log(OR)$ estimates, respectively.

For CDE parameters a term reflecting the contribution of estimating the conditional distribution of Z given A and W is incorporated into each influence curve, along with any dependence of missingness on Z:

$$\begin{aligned}
IC^{EY_1}(O) &= \frac{I(Z=z)}{g_{0_Z}(Z \mid W)} \frac{\Delta}{g_{0_\Delta}(1 \mid Z, W)} (Y - \bar{Q}_0(W)) + \bar{Q}_0(W) - \psi_0^{EY_1} \\
IC^{ATE}(O) &= \frac{I(Z=z)}{g_{0_Z}(Z \mid A, W)} \left(\frac{A}{g_{0_A}(1 \mid W)} - \frac{1-A}{g_{0_A}(0 \mid W)} \right) \frac{\Delta}{g_{0_\Delta}(1 \mid Z, A, W)} (Y - \bar{Q}_0(A, W)) \\
&\quad + \bar{Q}_0(1, W) - \bar{Q}_0(A, W) - \psi_0^{ATE} \\
IC^{logRR}(O) &= \frac{1}{\mu_1} \left(\frac{I(Z=z)}{g_{0_Z}(Z \mid A, W)} \frac{A}{g_{0_A}(1 \mid W)} \frac{\Delta}{g_{0_\Delta}(1 \mid Z, A, W)} (Y - \bar{Q}_0(A, W)) + \bar{Q}_0(1, W) - \mu_1 \right) \\
&\quad - \frac{1}{\mu_0} \left(\frac{I(Z=z)}{g_{0_Z}(Z \mid A, W)} \frac{1-A}{1 - g_{0_A}(1 \mid W)} \frac{\Delta}{g_{0_\Delta}(1 \mid Z, A, W)} (Y - \bar{Q}_0(A, W)) + \bar{Q}_0(0, W) - \mu_0 \right) \\
IC^{logOR}(O) &= \frac{1}{\mu_1(1-\mu_1)} \left(\frac{I(Z=z)}{g_{0_Z}(Z \mid A, W)} \frac{A}{g_{0_A}(1 \mid W)} \frac{\Delta}{g_{0_\Delta}(1 \mid Z, A, W)} (Y - \bar{Q}_0(A, W)) + \bar{Q}_0(1, W) \right) \\
&\quad - \frac{1}{\mu_0(1-\mu_0)} \left(\frac{I(Z=z)}{g_{0_Z}(Z \mid A, W)} \frac{1-A}{1 - g_{0_A}(1 \mid W)} \frac{\Delta}{g_{0_\Delta}(1 \mid Z, A, W)} (Y - \bar{Q}_0(A, W)) + \bar{Q}_0(0, W) \right)
\end{aligned}$$

Affiliation:

Susan Gruber
Harvard School of Public Health
Harvard University
E-mail: sgruber@hsph.harvard.edu
URL: http://works.bepress.com/sgruber/

Mark van der Laan
Division of Biostatistics
University of California, Berkeley
E-mail: laan@berkeley.edu
URL: http://www.stat.berkeley.edu/~laan/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 51, Issue 13 *Submitted:* 2011-02-07
November 2012 *Accepted:* 2012-10-02