



Dynamic Treatment Regimen Estimation via Regression-Based Techniques: Introducing R Package **DTRreg**

Michael P. Wallace
University of Waterloo

Erica E. M. Moodie
McGill University

David A. Stephens
McGill University

Abstract

Personalized medicine, whereby treatments are tailored to a specific patient rather than a general disease or condition, is an area of growing interest in the fields of biostatistics, epidemiology, and beyond. Dynamic treatment regimens (DTRs) are an integral part of this framework, allowing for personalized treatment of patients with long-term conditions while accounting for both their present circumstances and medical history. The identification of the optimal DTR in any given context, however, is a non-trivial problem, and so specialized methodologies have been developed for that purpose. Here we introduce the R package **DTRreg** which implements two regression-based approaches: G-estimation and dynamic weighted ordinary least squares regression. We outline the theory underlying these methods, discuss the implementation of **DTRreg** and demonstrate its use with hypothetical and real-world inspired simulated datasets.

Keywords: adaptive treatment strategies, dynamic treatment regimens, dynamic weighted ordinary least squares, G-estimation, personalized medicine.

1. Introduction

Dynamic treatment regimens (DTRs) are decision rules which take patient information such as current disease severity and treatment history as input and output a recommended treatment (Chakraborty and Moodie 2013; Chakraborty and Murphy 2014; Zhao and Laber 2014). At its simplest a DTR could be a single treatment decision of the form ‘prescribe treatment if patient is over 65 years old, prescribe control otherwise’ but they can also apply to longitudinal data contexts with multiple decision points. In this context patients can be given personalized management plans which evolve over time, taking into account their current condition, their medical history, and even their reactions to previous treatments. As such, DTRs can

lead to improvements in expected long-term outcomes when compared with one-size-fits-all strategies.

The problem of estimating the *optimal* DTR is of obvious interest, but brings with it theoretical challenges. In particular, the potential for past treatments to influence the impact of future ones requires careful consideration, as do the potentially complex model structures that can arise from applying sequences of decision rules. This has given rise to a rapidly expanding literature on the subject with a wide variety of new methods being proposed. As is typical for such a nascent area, however, such advances have focused largely on theoretical development rather than accessibility, with many methods presenting the familiar conflict of desirable theoretical properties but challenging implementation.

There are, of course, some notable exceptions. Q-learning (Watkins 1989; Sutton and Andrew 1998; Moodie, Dean, and Sun 2013), a relatively intuitive regression-based approach, is fairly accessible in itself but has also enjoyed computational implementation via the R package **qLearn** (Xin, Chakraborty, and Laber 2012). More recently, the alternative ‘interactive Q-learning’ approach (Laber, Linn, and Stefanski 2014) has received similar attention through the **iqLearn** (Linn, Laber, and Stefanski 2015a,b) package. Some more complex, classification-based approaches, meanwhile (Zhang, Tsiatis, Laber, and Davidian 2012, 2013; Zhao, Zeng, Rush, and Kosorok 2012; Zhao, Zeng, Laber, and Kosorok 2014), have similarly benefited from the release of R code in the form of the **DynTxRegime** package (Davidian *et al.* 2014). An important exception, however, is the estimating equations-based method of G-estimation (Robins 2004), which is notable for offering (as some of the aforementioned classification-based methods also do) model parameter estimates that are *doubly robust*. We shall discuss this property in more detail in the following section; in brief G-estimation delivers consistent parameter estimates as long as at least one of two models are correctly specified, thereby offering flexibility and robustness that is lacking in many simpler methods such as Q-learning. A related approach is dynamic weighted ordinary least squares (dWOLS; Wallace and Moodie 2015), a new method which borrows ideas from both Q-learning and G-estimation. dWOLS shares the double robustness property of G-estimation, but offers a simplicity of theory and implementation more akin to that which heightens the appeal of Q-learning (Wallace and Moodie 2015). In this paper we introduce the R package **DTRreg** (Wallace, Moodie, Stephens, and Simoneau 2017) which implements G-estimation, dWOLS and Q-learning for a variety of settings. The package is available from the Comprehensive R Archive Network (CRAN) at <https://CRAN.R-project.org/package=DTRreg>. We provide an outline of the theory underlying G-estimation and dWOLS in the sections that follow, but refer the interested reader to the associated methods papers (Robins 2004 and Wallace and Moodie 2015, respectively) for more detail.

2. G-estimation and dWOLS

2.1. Introductory concepts and notation

Much of the notation and terminology in the DTR literature will be familiar, but there are various concepts that require explicit introduction. In general we assume that individuals are followed up in stages: Starting at the beginning of treatment and at fixed points thereafter a treatment decision is taken based on currently available patient information. For example, a

patient prescribed an initial dose of some drug may then at weekly follow-up appointments have their prescription varied depending on the current severity of their condition. A DTR is simply the sequence of such treatment decision rules applied to that particular patient. While there is inevitably some variation in the literature, we shall use the following notation:

- y : Patient outcome, typically defined such that larger values are preferred.
- a_j : The j th treatment decision.
- \mathbf{h}_j : Covariate matrix containing patient information (*history*) prior to the j th treatment decision. Note that the history may include previous treatments a_1, \dots, a_{j-1} .

In general a subscript j is used to indicate the stage of treatment, of which there are typically J in total. A subscript i , when necessary, is used to indicate data on a specific patient, with n the total number of subjects under consideration. When using G-estimation, treatments may be binary, discrete, or continuous, while for dWOLS only binary treatments are supported. In this context we shall assume treatments are coded as either 1 or 0, which is often interpreted as representing receiving a treatment or a control, respectively. Finally, over- and underline notation is sometimes used to indicate the past and future. For example, \bar{a}_j can be used to indicate the first j treatment decisions, while \underline{a}_j would indicate treatments from the j th stage onward (up to the J th treatment).

A concept integral to DTR estimation is the *blip* function. Denoted $\gamma_j(\mathbf{h}_j, a_j)$, this is the difference in expected outcome between a patient who received treatment a_j at stage j and a patient with identical history who received some ‘reference’ treatment at stage j , assuming both go on to receive optimal future treatment. As such, this gives a measure of the impact of choosing treatment a_j instead of ‘standard’ care at stage j (i.e., the reference treatment), and so at each stage the optimal treatment is that which maximizes γ_j (where the subscript j notation reflects that these functions will typically vary from stage to stage).

If we consider a simple example where we have a single patient covariate and only one treatment decision is made (and thus suppress our subscript j notation), we could model our expected outcome as

$$\mathbb{E}[Y^a | \mathbf{H} = \mathbf{h}] = f(\mathbf{h}; \beta) + \gamma(\mathbf{h}, a; \psi), \quad (1)$$

where we have separated the model into two components: the blip function γ and the *treatment-free* function f . (The notation Y^a indicates a potentially unobserved, or *counterfactual* outcome Y under treatment a .) By compartmentalizing our outcome model in this way, it becomes clear that we need only focus on identifying the form and parameters of γ as this is the only means by which treatment can influence the expected outcome. Estimation of the blip function parameters ψ is therefore the focus of our DTR estimation endeavors. Note that we assume all blip functions will be linear in ψ .

Related to the blip is the *regret* function (Murphy 2003). This is defined as $\mu_j(\mathbf{h}_j, a_j) = \mathbb{E}[Y^{\bar{a}_{j-1}, \underline{a}_j^{\text{opt}}} - Y^{\bar{a}_j, \underline{a}_{j+1}^{\text{opt}}} | \mathbf{H}_j = \mathbf{h}_j]$ and gives the expected loss (or regret) in our outcome from using treatment a_j at stage j instead of the optimal treatment, assuming optimal treatment thereafter. Somewhat less formally, while the blip can be viewed as how ‘good’ it is to use a_j instead of some reference (control) treatment, the regret can be viewed as how ‘bad’ it is to use a_j instead of the best possible treatment at that stage. This relationship allows us to compute the regret from the blip via $\mu_j(\mathbf{h}_j, a_j) = \gamma_j(\mathbf{h}_j, a_j^{\text{opt}}) - \gamma_j(\mathbf{h}_j, a_j)$.

2.2. G-estimation

The G-estimation approach, like some other DTR estimation methods such as Q-learning and dWOLS, proceeds in a recursive manner. It begins by estimating the optimal final-stage treatment and then, using this information, moves backwards through stages estimating each previous optimal decision rule in turn. This recursive approach allows each subsequent decision to be based on the assumption of all future treatments being optimal. G-estimation's typical presentation is moderately intimidating (see Wallace and Moodie 2015 for an example), and we propose a more straightforward approach.

G-estimation for a binary treatment is implemented by the following process at each stage:

1. If $j = J$, write $\tilde{y}_J = y$. Otherwise, using results from earlier stages, form the *pseudo-outcome* $\tilde{y}_j = y + \sum_{k=j+1}^J \mu(\mathbf{h}_k, a_k; \hat{\psi}_k)$. This may be thought of as the expected outcome assuming optimal treatments are followed from stage $j + 1$ onward.
2. Propose a model of the form $\mathbb{E}[\tilde{Y}_j | \mathbf{h}_j, a_j; \beta_j, \psi_j] = \mathbf{h}_j^\beta \beta_j + a_j \mathbf{h}_j^\psi \psi_j$, where $\mathbf{h}_j^\beta \beta_j$ is our stage j *treatment-free* model, and $a_j \mathbf{h}_j^\psi \psi_j$ is our stage j *blip* model.
3. Propose a *treatment model* $\mathbb{E}[A_j | \mathbf{H}_j = \mathbf{h}_j; \alpha_j]$; use the data to obtain estimates $\hat{\alpha}_j$.
4. Form the matrices $\mathbf{h}_j^\delta = (\mathbf{h}_j^\beta, a_j \mathbf{h}_j^\psi)$ and $\mathbf{h}_j^w = (\mathbf{h}_j^\beta, \mathbf{h}_j^\psi (a_j - \mathbb{E}[A_j | \mathbf{h}_j^\alpha; \hat{\alpha}_j]))$, and estimate $\delta_j = (\beta_j, \psi_j)$ via $\hat{\delta}_j = [(\mathbf{h}_j^w)^\top \mathbf{h}_j^\delta]^{-1} (\mathbf{h}_j^w)^\top \tilde{y}_j$.

The resulting blip parameter estimators are then consistent if at least one of the treatment or treatment-free models is correctly specified (this being the double robustness property previously introduced). This is in contrast to estimators derived from methods such as Q-learning whose consistency relies on a single model being correct.

When treatment is a continuous, rather than binary, variable we may wish to modify our approach slightly. In particular, we may include a quadratic term in a_j in our blip function, thereby allowing it to potentially be maximized within the range of a_j rather than at one of its extremes (as would necessarily be the case if our blip were linear). Writing $\psi_j = (\psi_{j(1)}, \psi_{j(2)})$ and our quadratic blip function as $\gamma(\mathbf{h}_j, a_j; \psi_j) = a_j \mathbf{h}_j^{\psi(1)} \psi_{j(1)} + a_j^2 \mathbf{h}_j^{\psi(2)} \psi_{j(2)}$ then G-estimation can also be adapted to this case if we redefine our matrices as

$$\begin{aligned} \mathbf{h}_j^\delta &= (\mathbf{h}_j^\beta, a_j \mathbf{h}_j^{\psi(1)}, a_j^2 \mathbf{h}_j^{\psi(2)}); & \text{and} \\ \mathbf{h}_j^w &= (\mathbf{h}_j^\beta, \mathbf{h}_j^{\psi(1)} (a_j - \mathbb{E}[A_j | \mathbf{h}_j^\alpha; \hat{\alpha}_j]), \mathbf{h}_j^{\psi(2)} (a_j^2 - \mathbb{E}[A_j^2 | \mathbf{h}_j^\alpha; \hat{\alpha}_j])). \end{aligned}$$

2.3. Dynamic weighted ordinary least squares

The dynamic weighted ordinary least squares (dWOLS) method (Wallace and Moodie 2015) has much in common with G-estimation, but also builds on the theory (and simplicity) of Q-learning. Also a recursive approach, it involves conducting a sequence of weighted ordinary least squares regressions from which blip parameter estimates (and thence optimal DTRs) are derived. Again, we assume that the blip function at each stage is linear in ψ_j , while in addition we assume treatment is binary. Each stage of analysis involves the following steps

1. If $j = J$, write $\tilde{y}_J = y$. Otherwise, using results from earlier stages, form the *pseudo-outcome* $\tilde{y}_j = y + \sum_{k=j+1}^J \mu(\mathbf{h}_k, a_k; \hat{\psi}_k)$.
2. Propose a treatment model $E[A_j | \mathbf{H}_j = \mathbf{h}_j; \alpha_j]$; use the data to obtain estimates $\hat{\alpha}_j$.
3. Choose a weight function $w_j(a_j, \mathbf{h}_j; \alpha_j)$ which satisfies

$$\pi(\mathbf{h}_j)w_j(1, \mathbf{h}_j; \alpha_j) = (1 - \pi(\mathbf{h}_j))w_j(0, \mathbf{h}_j; \alpha_j),$$

where $\pi(\mathbf{h}_j) = P(A_j = 1 | \mathbf{h}_j)$. Use the estimates $\hat{\alpha}_j$ to obtain estimated weights \hat{w}_j .

4. Carry out a WOLS regression of \tilde{y}_j on $(\mathbf{h}_j^\beta, a_j \mathbf{h}_j^\psi)$ with weights \hat{w}_j where \mathbf{h}_j^β and \mathbf{h}_j^ψ are variables (or functions of variables) thought to be present in linear models of the treatment-free and blip functions. Use the resulting estimates $\hat{\psi}_j$ to construct the next pseudo-outcome and repeat the above steps, if necessary.

Like G-estimation, this produces consistent blip parameter estimators if at least one of the treatment or treatment-free models is correctly specified. In addition, it achieves this via a simple sequence of weighted regressions with relatively little algebraic or pre-computational clutter. In informal terms, this approach works because the ‘weighted dataset’ created by using weights of the given form removes any dependence between covariates and treatment when the treatment-free model is mis-specified.

Note that an entire family of weights can be used, and that the choice of weights can affect the efficiency of the resulting estimates. The theoretical underpinnings of choosing optimal weights are still under development, but so far we have found that weights of the form $w = |a - E[A | \mathbf{H}]|$ typically work well. These are the default weights used in the package, but should future work identify better weighting options we anticipate updating the routine accordingly, while in the meantime the user may choose to specify their own weights. Finally, we observe that if we perform the above steps but carry out a non-weighted regression (i.e., ordinary least squares rather than weighted) we are in fact performing Q-learning.

3. Implementation

The main function in **DTRreg** is `DTRreg`. The syntax is:

```
DTRreg(outcome, blip.mod, treat.mod, tf.mod, data = NULL,
       method = "gest", weight = "default", var.estim = "none", B = 200, M = 0,
       truncate = 0, verbose = FALSE, interrupt = FALSE, treat.range = NULL,
       missing = "default", interactive = FALSE, treat.mod.man = NULL)
```

The mandatory arguments are `outcome`, `blip.mod`, `treat.mod` and `tf.mod`. These define the various component models of the analysis, as well as the outcome variable and the dataset under analysis. `blip.mod`, `treat.mod` and `tf.mod` are lists of formulae which define the blip, treatment, and treatment-free models respectively at each stage, with the first element of each list corresponding to the first stage of treatment, the second element to the second stage, and so on. These also implicitly tell **DTRreg** how many stages are being considered (inferred from

the number of elements in each list, which must be the same for all three). If variables are contained in a data frame, this may be specified using `data`.

When treatment is binary and the true treatment model is known (as may be the case, for example, in a randomized trial) then this may be specified by setting `treat.mod.man` as a list of vectors which gives the probability of receiving treatment ($\mathbf{A} = 1$) for each subject at each stage. Note that in this case a dummy model for treatment should still be specified (such as `treat.mod = list(A1 ~ 1, A2 ~ 1)`) to indicate which are the treatment variables. If `treat.mod.man` is not set then the formulae specified in `treat.mod` are used in either a logistic or linear regression model for, respectively, a binary or non-binary treatment (which is inferred from the data).

Choice of analysis method (G-estimation, dWOLS, or Q-learning) is controlled by `method` which can be "gest", "dwols", or "qlearn". While the weights used for dWOLS are $\mathbf{w} = |\mathbf{a} - E[\mathbf{A}|\mathbf{H}]|$ by default, the `weight` option allows the user to specify a function in terms of $P(\mathbf{A} = 1|\mathbf{H})$ which will be used as the weights for all subjects with $a = 1$. The routine will then automatically assign weights for the remaining subjects to satisfy the condition in the dWOLS procedure outlined above. Selecting the method "qlearn" will cause the routines used for dWOLS to be called, but all regressions will be unweighted (by setting all weights equal to 1). This option is included primarily with exploratory analyses in mind, and is not intended to supersede or replace the extant **qLearn** package which offers some features that **DTRreg** does not, but is restricted to a simpler, two-interval, binary treatment setting. Note that while Q-learning does not require a treatment model to be specified, it is still necessary to identify the treatment variable(s). This may be done via the `treat.mod` option, where the model itself may be arbitrarily specified.

If desired, covariance estimates of the blip parameters can be estimated via bootstrap (by `var.estim = "bootstrap"`). When using G-estimation, meanwhile, an additional method based on the theory of [Robins \(2004\)](#) and [Moodie \(2009\)](#) is available (by `var.estim = "sandwich"`). The number of bootstrap resamples is controlled by `B`, while the m -out-of- n bootstrap may be used by specification of `M` for the bootstrap sample size (which otherwise defaults to n). If the variance is estimated via bootstrap then the arguments `truncate`, `verbose` and `interrupt` may be used. `truncate` allows the specification of a proportion (between zero and one) of the most extreme bootstrap parameter estimates which are excluded when covariance estimates are calculated. For example, setting `truncate = 0.01` removes the lowest 1% and highest 1% of bootstrapped parameter estimates prior to estimation of the standard error. This can be useful if the standard bootstrap procedure results in very large standard errors caused by a (relatively) small number of extreme estimates, although it is ad hoc and should only be used with considerable caution (it is recommended primarily for exploratory or sensitivity analysis purposes). `verbose` and `interrupt`, meanwhile, are useful for monitoring potentially time-intensive bootstrap analyses when conducting ad hoc, single analysis runs. The former produces estimated time to completion statistics while the latter gives the option to abort should the estimated time exceed 10 minutes.

The option `treat.range` is used when continuous treatments are being investigated and specifies the minimum and maximum possible values a treatment may take. By default these are taken to be the minimum and maximum values of the observed treatments at each stage of analysis. **DTRreg** will automatically ignore any subjects with missing data (thereby carrying out a complete-cases analysis), but if `missing = "ipcw"` then inverse probability of censored weights is used with the probability of censoring estimated via logistic regression on the full

covariate history up to that point. Finally, the `interactive = TRUE` option allows the user to enter variables to be used in the blip, treatment, and treatment-free models via on-screen, interactive prompts.

The `DTRreg` output includes blip parameter estimates along with standard errors and (Wald-type) confidence intervals if estimated. It is supported by `print()` (and `summary()`) functions which summarize the results of the analysis in an easy-to-understand format, including the stage-by-stage decision rules, while `coef()` and `confint()` may be used to extract information about parameter estimates. `predict()` is also available to provide estimated optimal outcomes for new data, but this assumes that the treatment-free models (along with the blip models) used were correctly specified. After analysis a `plot()` call produces diagnostic plots akin to those available following the use of standard regression commands such as `lm`. These include the usual diagnostics for the treatment model and specialized diagnostics that assess whether at least one of the treatment-free or blip models are incorrectly specified as per Rich, Moodie, Stephens, and Platt (2010).

4. Examples

4.1. Simulated example: dWOLS and binary treatment

We first demonstrate using dynamic weighted ordinary least squares for estimating the optimal DTR in a two-stage simulated dataset. Data are generated on $n = 10,000$ subjects as follows:

- Covariates: $X_j \sim N(0, 1)$.
- Treatments: $A_j \sim \text{Bin}(1, \text{expit}(x_j))$.
- Blip functions: $\gamma_j(x_j, a_j; \psi_j) = a_j(\psi_{j0} + \psi_{j1}x_j)$.
- Outcome: $Y \sim N(e^{x_1} + e^{x_2} + \gamma_1(x_1, a_1; \psi_1) + \gamma_2(x_2, a_2; \psi_2), 1)$.

$\text{expit}(x) = (1 + e^{-x})^{-1}$ is the expit or inverse-logit function. Setting $\psi_{j0} = \psi_{j1} = 1$ for $j = 1, 2$ we generate the above via the following R code:

```
R> set.seed(1)
R> expit <- function(x) 1 / (1 + exp(-x))
R> n <- 10000
R> X1 <- rnorm(n)
R> A1 <- rbinom(n, 1, expit(X1))
R> X2 <- rnorm(n)
R> A2 <- rbinom(n, 1, expit(X2))
R> gamma1 <- A1 * (1 + X1)
R> gamma2 <- A2 * (1 + X2)
R> Y <- exp(X1) + exp(X2) + gamma1 + gamma2 + rnorm(n)
```

We can then conduct a dWOLS analysis on these data using the `DTRreg` command with the `method = "dwols"` option. Here we present an example output for illustration, where we have specified a non-default weight function: the inverse probability of treatment weights

(IPTW), defined as $w_i = P(A_i = a_i | X_i = x_i)^{-1}$ (that is, the reciprocal of the probability a patient received their given treatment). Because these weights satisfy the condition in Step 3 of our dWOLS procedure above we may use them for our analysis. As noted above, we use the `weight` option and define our weight function in terms of $E[A|X] = P(A = 1|X = x)$ for those subjects with $a = 1$. For the IPTW this corresponds to the function $f(w) = \frac{1}{w}$: For subjects with $a = 1$ the weight is simply $E[A|X]^{-1}$ while the weight for subjects with $a = 0$ is constructed to satisfy the dWOLS condition (and as such equates to $(1 - E[A|X])^{-1}$). First, we specify the models to be passed to `DTRreg`:

```
R> blip.mod <- list(~ X1, ~ X2)
R> treat.mod <- list(A1 ~ X1, A2 ~ X2)
R> tf.mod <- list(~ X1, ~ X2)
```

We then specify our weight function:

```
R> weight.fun <- function(w) 1 / w
```

Finally, we can run our analysis, requesting bootstrapped standard errors (and confidence intervals):

```
R> DTRreg(Y, blip.mod, treat.mod, tf.mod, method = "dwols",
+   weight = weight.fun, var.estim = "bootstrap")
```

DTR estimation over 2 stages:

Blip parameter estimates

	Estimate	Std. Error	95% Conf. Int
Stage 1 (n = 10000)			
(Intercept)	1.1064	0.0775	[0.9545,1.2584]
X1	1.1524	0.0918	[0.9724,1.3324]
Stage 2 (n = 10000)			
(Intercept)	1.1362	0.0638	[1.0111,1.2613]
X2	1.1186	0.0981	[0.9263,1.3109]

Warning: possible non-regularity at stage 1 (prop = 0.134)

Warning: possible non-regularity at stage 2 (prop = 0.131)

Recommended dynamic treatment regimen:

Stage 1: treat if $1.1064 + 1.1524 X1 > 0$

Stage 2: treat if $1.1362 + 1.1186 X2 > 0$

Much of this output is self-explanatory: the blip parameter estimates are $(\psi_{10}, \psi_{11}) = (1.1064, 1.1524)$ and $(\psi_{20}, \psi_{21}) = (1.1362, 1.1186)$ at stages 1 and 2, respectively. These give rise to the recommended treatment regimen detailed at the end of the output, providing a straightforward summary of how the analysis translates to a DTR.

This example also demonstrates non-regularity warnings. In this context, non-regularity occurs if the optimal treatment for a given patient is non-unique. Non-regularity can cause

theoretical concerns for both G-estimation and dWOLS, and so the commands in **DTRreg** carry out an informal check based on the data and estimated blip parameters (see page 317 of [Robins 2004](#), for further details). This involves identifying the proportion of subjects for whom, when all possible blip parameter values within their respective confidence sets are considered, both treatment and non-treatment could be recommended. In the example above the proportion of subjects for whom this was the case is 0.134 at stage 1 and 0.131 at stage 2, and so the output includes a warning to that effect. Because non-regularity can only be estimated when standard errors are, we note that **DTRreg** will not display non-regularity warnings if `var.estim = "none"`. The m -out-of- n bootstrap is one built-in option that can help with the problem of non-regularity, while for a more general discussion see Chapter 8 of [Chakraborty and Moodie \(2013\)](#).

Various information is stored in the model object, and we draw particular attention to the stored variable `opt.Y`. This contains the estimated outcome for each subject had they followed the optimal regimen, assuming the treatment-free and blip models were correctly specified, and that the resulting parameter estimates are correct. This can give some sense of the potential improvement in outcome had subjects followed the optimal regimen, although obviously it should be viewed with some caution given the underlying assumptions.

We also conducted four sets of longer simulation runs where neither, one, or both of the treatment and treatment-free models were correctly specified, using dWOLS as above along with G-estimation and Q-learning. The treatment-free model was mis-specified as in the above example, with terms in x_1 and x_2 rather than e^{x_1} and e^{x_2} , while the treatment model was mis-specified by assuming treatments were assigned with probability 0.5. The results of these larger simulation runs ([Table 1](#)) demonstrate the expected consistency of dWOLS and G-estimation when at least one of these two models is correctly specified, while Q-learning (which of course does not depend on the treatment-free model) exhibits considerable bias when the treatment-free model is mis-specified, similar to what is observed under dWOLS and G-estimation when both are mis-specified.

Finally, in [Figures 1](#) and [2](#) we illustrate partial results of calling the `plot()` command on a ‘**DTRreg**’ object (here, with a sample size of $n = 1,000$ for a slightly cleaner plot). The diagnostic plots presented demonstrate that when the treatment-free model is incorrectly specified there is evidence of a relationship between the residuals, fitted values and the model covariates. The `plot()` command also returns standard diagnostics for the treatment model, which are omitted here.

4.2. Simulated example: G-estimation and continuous treatment

Next we present a similar example but with a continuous, rather than binary, treatment. Again, we shall consider a two-stage setup, with datasets of size $n = 10,000$ generated as follows:

- Covariates: $X_j \sim N(0, 1)$.
- Treatments: $A_j \sim N(x_j, 0.5)$.
- Blip functions: $\gamma_j(x_j, a_j; \psi_j) = a_j(\psi_{j0} + \psi_{j1}x_j + \psi_{j2}a_j)$.
- Outcome: $Y \sim N(e^{x_1} + e^{x_2} + \gamma_1(x_1, a_1; \psi_1) + \gamma_2(x_2, a_2; \psi_2), 1)$.

Method	Models	Estimates			
		$\hat{\psi}_{10}$	$\hat{\psi}_{11}$	$\hat{\psi}_{20}$	$\hat{\psi}_{21}$
G-estimation	Both correct	1.001	0.999	1.002	1.003
	Treatment correct	1.001	0.998	1.002	1.004
	Treatment-free correct	1.001	1.002	1.001	1.004
	Neither correct	0.922	2.334	0.923	2.338
dWOLS	Both correct	1.001	0.999	1.002	1.002
	Treatment correct	1.001	1.006	1.003	1.009
	Treatment-free correct	1.001	0.999	1.003	1.000
	Neither correct	0.922	2.332	0.923	2.331
Q-learning	Treatment-free correct	1.003	0.999	0.999	1.004
	Treatment-free incorrect	0.926	2.337	0.921	2.337

Table 1: Mean blip parameter estimates and improvement in (mean) outcome from dWOLS, G-estimation, and Q-learning analysis of 1,000 simulated datasets of size $n = 10,000$ with neither, one, or both the treatment and treatment-free models correctly specified. Note that Q-learning does not depend on the choice of a treatment model.

For our example $\psi_{j0} = \psi_{j1} = 1$ and $\psi_{j2} = -1$ (for $j = 1, 2$). In addition to the treatments now depending linearly on the patient covariates the other substantial change from the binary setup is in the form of the blip. By including a quadratic term in a_j we allow for the possibility that the blip is maximized at some internal value of a_j rather than necessarily at its maximum or minimum. However, we note that linear blip functions are still permissible when treatments are continuous, with either the user-specified `treat.range` or one inferred from the data used for optimal rule estimation. Note that `DTRreg` can only be used for blip functions that are linear or quadratic in the treatment variable; no higher order terms are currently supported.

The above data can be generated in R using the following code:

```
R> set.seed(1)
R> n <- 10000
R> X1 <- rnorm(n)
R> A1 <- rnorm(n, X1, 0.5)
R> X2 <- rnorm(n)
R> A2 <- rnorm(n, X2, 0.5)
R> gamma1 <- A1 * (1 + X1 - A1)
R> gamma2 <- A2 * (1 + X2 - A2)
R> Y <- exp(X1) + exp(X2) + gamma1 + gamma2 + rnorm(n)
```

We illustrate the implementation of the `DTRreg` command, again making use of the bootstrap variance estimation procedure.

```
R> blip.mod <- list(~ X1 + A1, ~ X2 + A2)
R> treat.mod <- list(A1 ~ X1, A2 ~ X2)
R> tf.mod <- list(~ X1, ~ X1 + A1 + X2)
R> DTRreg(Y, blip.mod, treat.mod, tf.mod, var.estim = "bootstrap", B = 200,
+   treat.range = c(-5, 5))
```

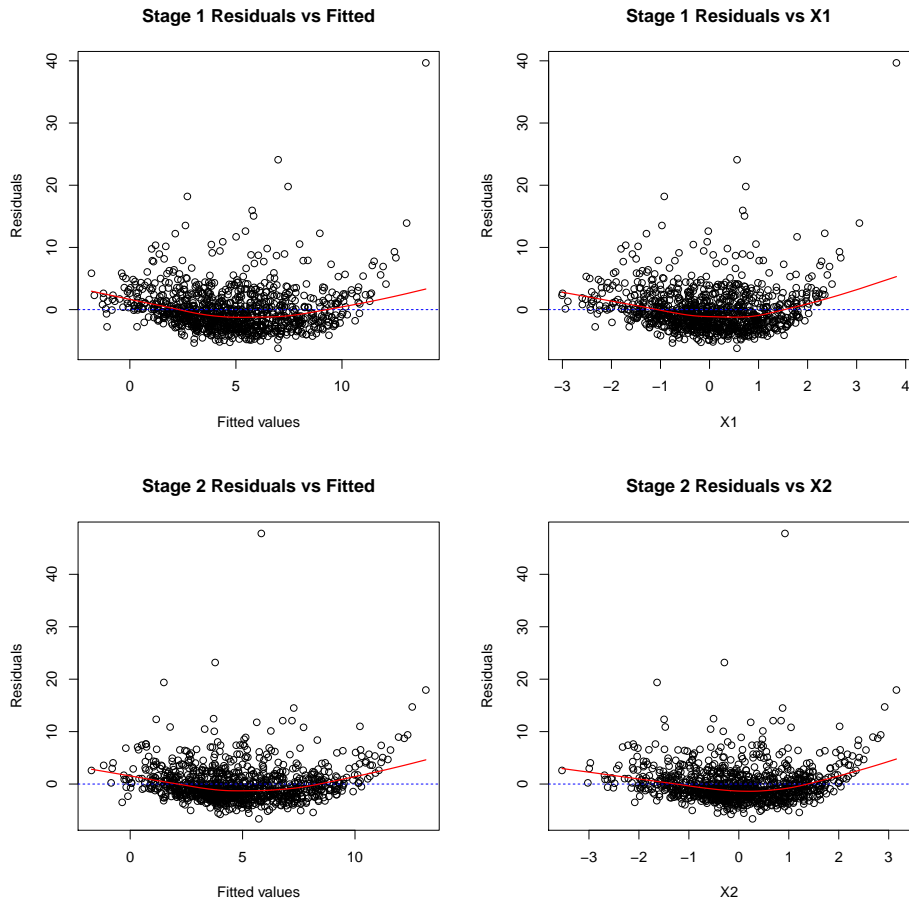


Figure 1: Diagnostic plots of treatment-free or blip function mis-specification when the former is incorrectly specified.

DTR estimation over 2 stages:

Blip parameter estimates

	Estimate	Std. Error	95% Conf. Int
Stage 1 (n = 10000)			
(Intercept)	1.0575	0.0601	[0.9397, 1.1754]
X1	1.2970	0.1629	[0.9776, 1.6163]
A1	-1.0959	0.0814	[-1.2555, -0.9363]
Stage 2 (n = 10000)			
(Intercept)	1.0659	0.0452	[0.9773, 1.1544]
X2	1.2231	0.1194	[0.9890, 1.4571]
A2	-1.0759	0.0599	[-1.1933, -0.9585]

The output is interpreted similarly to that of the binary treatment, dWOLS example above, with the addition of standard errors and 95% confidence intervals estimated via the bootstrap. A run of 1,000 such simulated datasets returns mean blip parameter estimates of $(\hat{\psi}_1, \hat{\psi}_2) =$

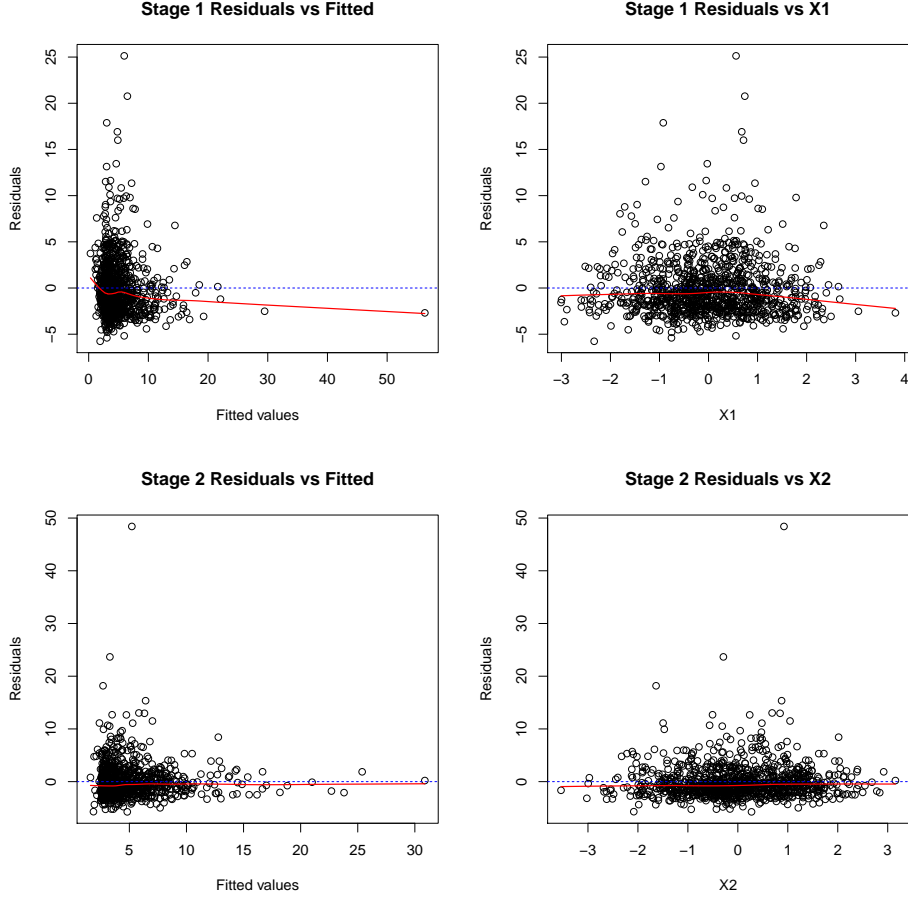


Figure 2: Diagnostic plots of treatment-free or blip function mis-specification when both are correctly specified.

(1.000, 0.986, -0.996, 0.998, 1.001, -1.002) reflecting the consistency we would expect despite the treatment-free model being mis-specified.

We note that with a quadratic blip function the output does not include a summary of the recommended DTR, as in this setting such rules are potentially quite complex. For clarity, we present a general interpretation of the blip parameters in this context.

Suppose our blip function at stage j is of the form $\gamma_j(x_j, a_j; \psi_j) = a_j(\psi_{j0} + \psi_{j1}x_j + \psi_{j2}a_j)$ and we obtain corresponding parameter estimates $(\hat{\psi}_{j0}, \hat{\psi}_{j1}, \hat{\psi}_{j2})$. Depending on whether $\hat{\psi}_{j2}$ is negative or positive, the blip is then maximized or minimized, respectively, for a specific subject by solving $\frac{\partial \hat{\gamma}_j}{\partial a_j} = 0$, giving $a_j^* = \frac{-(\hat{\psi}_{j0} + \hat{\psi}_{j1}x_j)}{2\hat{\psi}_{j2}}$. We would then recommend treatment a_j^* if it lies within the range of possible values for a_j (and is a maximum rather than a minimum). Otherwise the optimal treatment will be that which of the maximum or minimum possible value for a_j maximizes the estimated blip.

4.3. The Promotion of Breastfeeding Intervention Trial

As our final illustration we work with a simulated dataset that is inspired by real data. The

Promotion of Breastfeeding Intervention Trial (PROBIT; Kramer *et al.* 2001) randomized mother-infant pairs across 31 Belarusian maternity hospitals to receive either standard care or a breastfeeding encouragement intervention. We follow Chakraborty and Moodie (2013) and Moodie, Chakraborty, and Kramer (2012) in investigating the effect of breastfeeding from 0–3 months and 3–6 months on a child’s later development. This is measured by administration of the Wechsler abbreviated scales of intelligence when the child is 6.5 years of age, encompassing tests of verbal and quantitative intelligence. (We note that application of dWOLS to the real dataset is presented in Wallace and Moodie 2015.)

We consider investigating the effect of breastfeeding on an outcome of verbal IQ in a two-stage setup (Kramer *et al.* 2008). Stage 1 treatment is defined as whether any breastfeeding took place between birth and three months, while stage 2 treatment is defined as whether any breastfeeding took place between three and six months. Our simulated outcome is formed of a linear combination of baseline covariates (the treatment-free model) along with two blip functions – one for each stage of treatment. The baseline covariates are intervention group status, hospital location (East or West Belarus, and urban or rural), mother’s education level (a three-value categorical variable), mother’s smoking status, family history of allergy, mother’s age, mother’s breastfeeding of previous children, whether the birth was by cesarean section, the infant’s sex and birth weight. For more details on the simulation details see Appendix A. Our blip functions take the form

- $\gamma_1 = a_1(-0.2 + 0.1 \times \text{Weight}_0)$; and
- $\gamma_2 = a_2(-0.65 + 0.1 \times \text{Weight}_3 - 0.5 \times \text{Infection} - 0.5 \times \text{Hospitalization})$,

where Weight_0 and Weight_3 denote the infant’s weight at zero and three months, respectively, while Infection and Hospitalization are binary variables taking value 1 if the infant was reported as having had an infection, or been hospitalized, during the first three months. Intervention status, age, location and infections during the first three months were independently generated based on their observed distribution in the PROBIT dataset. Other variables were generated with some dependencies (largely on education level) based on exploratory analyses of the data. Breastfeeding, weight at three months, and our two outcomes were generated based on parameters derived from regression models using the full dataset. Stage 1 treatment was modeled via logistic regression on intervention status, mother’s education, mother’s age, mother’s breastfeeding of previous children and the infant’s sex. Stage 2 treatment was similar, but with stage 1 treatment included as an extra covariate. Weight at three months was based on a linear regression on the full set of baseline covariates and an interaction term between breastfeeding during the first three months and birth weight. Finally, our outcome was generated as a linear combination of the baseline covariates (with parameters estimated via linear regression) and the blip functions above.

The blip parameters chosen for the above outcome models correspond to a regime that during the first three months recommends breastfeeding for all infants whose birth weight exceeds 2 kilograms. Between three and six months an infant should be breastfed if their three month weight exceeds 6.5 kilograms if they did not have an infection or hospitalization during the first three months, 6.0 kilograms if they had either an infection or hospitalization, and 5.5 kilograms if they had both. These parameter choices were inspired in part by the findings of Chakraborty and Moodie (2013) as well as to present a more complex stage 2 treatment rule. Note that our second stage parameters reflect an intuition that infants who were ill

or hospitalized during their first three months may be ‘artificially’ underweight and so the breastfeeding threshold is lower.

Despite this complex setup, analysis via G-estimation (or dWOLS) using **DTRreg** is straightforward. Code detailing the full setup of the dataset is not included here, but we present some sample output below. Note that we model the log-transformed quantitative IQ score, which we have called `verbal.iq`.

```
R> PROBIT.sim <- read.csv("Simulated_PROBIT_data.csv")
R> bl.mod <- list(~ Wgt0, ~ Wgt3 + Infc + Hosp)
R> tr.mod <- list(Brf1 ~ Intervention + factor(Educ) + Age + BF.prev +
+   BF.not + Sex, Brf2 ~ Brf1 + Intervention + factor(Educ) + Age +
+   BF.prev + BF.not + Sex)
R> trf.mod <- list(~ Intervention + factor(Location) + factor(Educ) +
+   Smoke + Allergy + Age + I(Age^2) + BF.prev + BF.not + Cesarean +
+   Sex + Wgt0, ~ Intervention + factor(Location) + factor(Educ) +
+   Smoke + Allergy + Age + I(Age^2) + BF.prev + BF.not + Cesarean +
+   Sex + Infc + Hosp + Wgt0 + Brf1 + Wgt3)
R> verb.mod <- DTRreg(verbal.iq, bl.mod, tr.mod, trf.mod, data = PROBIT.sim,
+   method = "gest", var.estim = "sandwich")
R> verb.mod
```

DTR estimation over 2 stages:

Blip parameter estimates

	Estimate	Std. Error	95% Conf. Int
Stage 1 (n = 17044)			
(Intercept)	-0.2023	0.0040	[-0.2102,-0.1945]
Wgt0	0.1012	0.0012	[0.0988,0.1035]
Stage 2 (n = 17044)			
(Intercept)	-0.6508	0.0010	[-0.6528,-0.6488]
Wgt3	0.1001	0.0002	[0.0998,0.1005]
Infc	-0.4999	0.0001	[-0.5002,-0.4997]
Hosp	-0.5001	0.0002	[-0.5004,-0.4997]

Recommended dynamic treatment regimen:

Stage 1: treat if $-0.2023 + 0.1012 \text{ Wgt0} > 0$

Stage 2: treat if $-0.6508 + 0.1001 \text{ Wgt3} - 0.4999 \text{ Infc} - 0.5001 \text{ Hosp} > 0$

5. Discussion

The **DTRreg** package provides an easy means by which to conduct DTR analyses via the established method of G-estimation and the more novel dWOLS approach. These methods offer the desirable double robustness property whereby correct specification of at least one of

the treatment or treatment-free models results in consistent parameter estimation. In practice, and particularly with trial data, the former of these may well be known precisely and thus these methods can be expected to perform well, but double robustness offers an additional flexibility to an analysis. By making these methods accessible through the R environment we hope many more practitioners will be encouraged to incorporate them in their work.

As with any computational package we anticipate improving and updating the code as theory develops and time allows. Of particular interest would be expanding to other data structures, such as non-continuous outcomes and higher order treatment terms in blip functions. As the corpus of computational routines for methods from the DTR literature expands, we hope to see the emergence of a coherent suite for dynamic treatment regimen analysis.

Acknowledgments

This work is funded by Discovery Grants from the Natural Sciences and Engineering Research Council (NSERC) of Canada. Dr. Moodie is sponsored by a Chercheur-boursier career award from the Fonds de recherche du Québec-Santé (FRSQ). The authors would like to thank Michael Kramer for allowing us to use the PROBIT data. The Promotion of Breastfeeding Intervention Trial (PROBIT) was supported by grants from the Thrasher Research Fund, the National Health Research and Development Program (Health Canada), UNICEF, the European Regional Office of WHO, and the CIHR.

References

- Chakraborty B, Moodie EEM (2013). *Statistical Methods for Dynamic Treatment Regimes*. Springer-Verlag. doi:10.1007/978-1-4614-7428-9.
- Chakraborty B, Murphy SA (2014). “Dynamic Treatment Regimes.” *Annual Review of Statistics and Its Application*, **1**, 447–464. doi:10.1146/annurev-statistics-022513-115553.
- Davidian M, Laber EB, Linn KA, Stefanski LA, Tsiatis AA, Zhang B, Zhang M, Holloway ST (2014). **DynTxRegime: Methods for Estimating Dynamic Treatment Regimes**. R package version 2.1, URL <https://CRAN.R-project.org/package=DynTxRegime>.
- Kramer MS, Aboud F, Mironova E, Vanilovich I, Platt R, Matush L, Igunnov S, Fombonne E, Bogdanovich N, Ducruet T, Collet J, Chalmers B, Hodnett E, Davidovsky S, Skugarevsky O, Trofimovich O, Kozlova L, Shapiro S (2008). “Breastfeeding and Child Cognitive Development: New Evidence from a Large Randomized Trial.” *Archives of General Psychiatry*, **65**(5), 578–584. doi:10.1001/archpsyc.65.5.578.
- Kramer MS, Chalmers B, Hodnett ED, Sevkovskaya Z, Dzikovich I, Shapiro S, Collet J, Vanilovich I, Mezen I, Ducruet T, Shishko G, Zubovich V, Mknuiik D, Gluchanina E, Dombrovsky V, Ustinovitch A, Ko T, Bogdanovich N, Ovchinikova L, Helsing E (2001). “Promotion of Breastfeeding Intervention Trial (PROBIT): A Randomized Trial in the Republic of Belarus.” *Journal of the American Medical Association*, **285**(4), 413–420. doi:10.1001/jama.285.4.413.

- Laber EB, Linn KA, Stefanski LA (2014). “Interactive Model Building for Q-Learning.” *Biometrika*, **101**(4), 831–847.
- Linn KA, Laber EB, Stefanski LA (2015a). **iqLearn**: *Interactive Q-Learning*. R package version 1.4, URL <https://CRAN.R-project.org/package=iqLearn>.
- Linn KA, Laber EB, Stefanski LA (2015b). “**iqLearn**: Interactive Q-Learning in R.” *Journal of Statistical Software*, **64**(1), 1–25. doi:10.18637/jss.v064.i01.
- Moodie E, Chakraborty B, Kramer M (2012). “Q-Learning for Estimating Optimal Dynamic Treatment Rules from Observational Data.” *Canadian Journal of Statistics*, **40**(4), 629–645. doi:10.1002/cjs.11162.
- Moodie EEM (2009). “A Note on the Variance of Doubly-Robust G-Estimators.” *Biometrika*, **96**(4), 998–1004.
- Moodie EEM, Dean N, Sun YR (2013). “Q-Learning: Flexible Learning about Useful Utilities.” *Statistics in Biosciences*, **6**(2), 223–243. doi:10.1007/s12561-013-9103-z.
- Murphy S (2003). “Optimal Dynamic Treatment Regimes.” *Journal of the Royal Statistical Society B*, **65**(2), 331–355. doi:10.1111/1467-9868.00389.
- Rich B, Moodie EEM, Stephens DA, Platt RW (2010). “Model Checking with Residuals for G-Estimation of Optimal Dynamic Treatment Regimes.” *International Journal of Biostatistics*, **6**(2), Article 12. doi:10.2202/1557-4679.1210.
- Robins JM (2004). “Optimal Structural Nested Models for Optimal Sequential Decisions.” In D Lin, P Heagerty (eds.), *Proceedings of the Second Seattle Symposium on Biostatistics*, pp. 189–326. Springer-Verlag.
- Sutton RS, Andrew GB (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge.
- Wallace M, Moodie EEM, Stephens DA, Simoneau G (2017). **DTRreg**: *DTR Estimation and Inference via G-Estimation, Dynamic WOLS, and Q-Learning*. R package version 1.2, URL <https://CRAN.R-project.org/package=DTRreg>.
- Wallace MP, Moodie EEM (2015). “Doubly-Robust Dynamic Treatment Regimen Estimation via Weighted Least Squares.” *Biometrics*, **71**(3), 636–644. doi:10.1111/biom.12306.
- Watkins CJCH (1989). *Learning from Delayed Rewards*. Cambridge University Press, Cambridge, England.
- Xin J, Chakraborty B, Laber EB (2012). **qLearn**: *Estimation and Inference for Q-Learning*. R package version 1.0, URL <https://CRAN.R-project.org/package=qLearn>.
- Zhang B, Tsiatis AA, Laber EB, Davidian M (2012). “A Robust Method for Estimating Optimal Treatment Regimes.” *Biometrics*, **68**(4), 1010–1018. doi:10.1111/j.1541-0420.2012.01763.x.
- Zhang B, Tsiatis AA, Laber EB, Davidian M (2013). “Robust Estimation of Optimal Dynamic Treatment Regimes for Sequential Treatment Decisions.” *Biometrika*, **100**(3), 681–694.

- Zhao Y, Laber EB (2014). “Estimation of Optimal Dynamic Treatment Regimes.” *Clinical Trials*, **11**(4), 400–407. doi:[10.1177/1740774514532570](https://doi.org/10.1177/1740774514532570).
- Zhao Y, Zeng D, Rush AJ, Kosorok MR (2012). “Estimating Individual Treatment Rules Using Outcome Weighted Learning.” *Journal of the American Statistical Association*, **107**(499), 1106–1118. doi:[10.1080/01621459.2012.695674](https://doi.org/10.1080/01621459.2012.695674).
- Zhao YQ, Zeng D, Laber EB, Kosorok MR (2014). “New Statistical Learning Methods for Estimating Optimal Dynamic Treatment Regimes.” *Journal of the American Statistical Association*, **110**(510), 583–598. doi:[10.1080/01621459.2014.937488](https://doi.org/10.1080/01621459.2014.937488).

A. PROBIT simulation details

Simulated data based on PROBIT. Note that ‘infection’ and ‘hospitalization’ refer to those events during stage 1. Here is a ‘quick’ summary of the generation structure:

- Independently generated variables: intervention status, age, location, infection.
- Dependent on location: education.
- Dependent on education: location, smoking status, history of allergy, whether the birth was by cesarean section, whether breastfed previously.
- Dependent on infection: hospitalization.
- Dependent on sex: weight.
- Dependent on breastfed previously: ‘did not breastfeed’.
- Breastfeeding during stages 1 and 2, and weight at 3 months were generated based on regression models using the real PROBIT data (using the variables used in the example in Section 6.1 of [Chakraborty and Moodie 2013](#)).
- IQ scores (verbal and quantitative) were generated by adding a ‘treatment free’ predicted value to blip functions for stages 1 and 2. The blip functions are:

$$\begin{aligned}\gamma_1 &= A_1(\psi_{10} + \psi_{20}\text{Weight}_0), \\ \gamma_2 &= A_2(\psi_{20} + \psi_{21}\text{Weight}_3 + \psi_{22}\text{Infection} + \psi_{23}\text{Hospitalization}),\end{aligned}$$

where we choose blip parameters as desired for the verbal IQ, and set them all to zero for the quantitative IQ. The treatment free value for each subject was generated based on a linear regression of observed vocabulary score on the variables used in the same example from the book, but when predicting using this model all treatments were assumed to be zero. In other words, if $\widehat{Y}(H_\beta, H_\psi; \boldsymbol{\beta}, \boldsymbol{\psi})$ is the predicted value from this linear model for a subject where we have divided the model covariates into those not including treatment terms (H_β) and those which do (H_ψ), then our simulated IQ scores are generated by $Y = \widehat{Y}(H_\beta, H_\psi = 0; \boldsymbol{\beta}, \boldsymbol{\psi}) + \gamma_1 + \gamma_2$.

Below are the precise details of the data generation. Specific numbers are derived from observations made on the original PROBIT dataset.

- Intervention status: intervention with probability 0.52.
- Age: log-normal with mean 3.17 and variance 0.037.
- Sex: male with probability 0.52.
- Location: assigned to urban East Belarus, rural East Belarus, urban West Belarus and rural West Belarus with probabilities (0.33, 0.16, 0.26, 0.26).

- Education: assigned to one of three levels dependent on location. Education levels 1, 2 and 3 correspond to ‘no university’, ‘some university’ and ‘completed university’ education, respectively.
 - East Belarus (urban): $P(\text{Education} = (1, 2, 3)) = (0.31, 0.54, 0.15)$.
 - East Belarus (rural): $P(\text{Education} = (1, 2, 3)) = (0.44, 0.44, 0.11)$.
 - West Belarus (urban): $P(\text{Education} = (1, 2, 3)) = (0.33, 0.51, 0.16)$.
 - West Belarus (rural): $P(\text{Education} = (1, 2, 3)) = (0.41, 0.48, 0.11)$.
- Smoking: $P(\text{Smoking} = 1 | \text{Education} = (1, 2, 3)) = (0.03, 0.02, 0.01)$.
- Allergy: $P(\text{Allergy} = 1 | \text{Education} = (1, 2, 3)) = (0.03, 0.05, 0.07)$.
- Cesarean: $P(\text{Cesarean} = 1 | \text{Education} = (1, 2, 3)) = (0.10, 0.12, 0.16)$.
- Had previous children and breastfed: $P(\text{BFPrev} = 1 | \text{Education} = (1, 2, 3)) = (0.25, 0.23, 0.30)$.
- Had previous children and did not breastfeed: with probability 0.23 if not previously breastfed (0 otherwise).
- Infection: with probability 0.19.
- Hospitalization: with probability 0.07 if no infection, 0.25 if infection.
- Birth weight: Normal with $(\mu, \sigma) = (3.36, 0.39)$ for females, $(3.5, 0.42)$ for males.
- Stage 1 breastfeeding: predicted using results of a logistic regression (using PROBIT data) of stage 1 breastfeeding on intervention, education, age, breastfed previously, did not breastfeed, and sex.
- 3 month weight: predicted using results of a linear regression model (using PROBIT data) of (log) weight at 3 months on intervention, location, education, smoking, allergy, age, age², breastfed previously, did not breastfeed, cesarean, sex, infection, hospitalization, birth weight, stage 1 breastfeeding, and an interaction term between stage 1 breastfeeding and birth weight.
- Stage 2 breastfeeding: predicted using results of a logistic regression (using PROBIT data) of stage 2 breastfeeding on stage 1 breastfeeding, intervention, education, age, previously breastfed, did not breastfeed, sex.
- Outcome (IQ scores): predicted using results of a linear regression (using PROBIT data) of all the terms in the 3 month weight model and: weight at 3 months, stage 2 breastfeeding, and interactions between stage 2 breastfeeding and weight at 3 months, and between stage 2 breastfeeding and stage 1 breastfeeding. IQ scores generated by predicting using this model with treatment variables ignored, then adding the appropriate blip functions for stages 1 and 2.

Affiliation:

Michael P Wallace
Department of Statistics and Actuarial Science
University of Waterloo
Waterloo, Canada
E-mail: michael.wallace@uwaterloo.ca