# rbw: An R package for Constructing Residual Balancing Weights

*by Derick Baum and Xiang Zhou*

**Abstract** We describe the R package **rbw**, which implements the method of residual balancing weights (RBW) for estimating marginal structural models. In contrast to other methods such as inverse probability weighting (IPW) and covariate balancing propensity scores (CBPS), RBW involves modeling the conditional means of post-treatment confounders instead of the conditional distributions of the treatment to construct the weights. RBW is thus easier to use with continuous treatments, and the method is less susceptible to model misspecification issues that often arise when modeling the conditional distributions of treatments. RBW is also advantageous from a computational perspective. Because its weighting procedure involves a convex optimization problem, RBW typically locates a solution considerably faster than other methods whose optimization relies on nonconvex loss functions — such as the recently proposed nonparametric version of CBPS. We explain the rationale behind RBW, describe the functions in **rbw**, and then use real-world data to illustrate their applications in three scenarios: effect estimation for point treatments, causal mediation analysis, and effect estimation for time-varying treatments with time-varying confounders.

## Introduction

This paper describes the R package **rbw**, which implements the method of residual balancing weights for estimating marginal structural models (MSMs) (Zhou and Wodtke, 2020). MSMs seek to estimate causal effects in the presence of post-treatment confounding — a common issue in the social sciences. In studies of time-varying treatments, prior treatments may affect the confounders of future treatments. For example, research has shown that political candidates' decision to run negative advertisements is shaped by their positions in recent polling data, which are in turn affected by their previous decisions to run negative advertisements (Lau et al., 2007; Blackwell, 2013). Post-treatment confounding can also occur in causal mediation analysis when confounders of the mediator-outcome relationship are affected by the treatment. For example, such a problem arises in a study of the mediating role of morality in the effect of shared democracy on public support for war (Tomz and Weeks, 2013). Post-treatment variables, such as respondents' beliefs about the likelihood of victory, may affect both perceptions of morality and support for military intervention.

MSMs aim to address two types of bias associated with conventional regression methods that adjust naively for post-treatment confounders: overcontrol and collider-stratification bias (Robins, 1986; 2000). Conditioning naively on post-treatment confounders can create overcontrol bias as it blocks the effect of the treatment on the outcome that passes through these variables. Additionally, conditioning naively on post-treatment confounders can lead to collider-stratification bias when these variables are affected by unobserved determinants of the outcome. This is because the adjustment will create a spurious association between the treatment and the unobserved variables.

Researchers often use inverse probability weighting (IPW) to fit MSMs (for an in-depth exposition of the method, see Robins et al., 2000; Robins, 2000; Cole and Hernán, 2008). In longitudinal settings, MSMs with IPW involve fitting a model for the conditional mean of the outcome given the treatment history using weights that break the dependence between past confounders and the treatment at each time point. In essence, the weights create a pseudo-population where the marginal mean of the potential outcomes under a treatment history equals the conditional mean of the observed outcome given the treatment history. The R package **ipw** provides functions for estimating inverse probability weights (van der Wal and Geskus, 2011; Geskus and van der Wal, 2015).

However, IPW's success depends on correct specification of the models for the conditional distributions of exposure to treatment and/or mediator (hereafter jointly referred to as "exposures"), which is difficult to achieve in practice. Moreover, even when these models are correctly specified, IPW is inefficient and susceptible to finite-sample bias (Zhou and Wodtke, 2020; Wang et al., 2006). Finally, when the exposures are continuous, IPW may perform poorly due to unreliable estimation of conditional densities (Naimi et al., 2014; Vansteelandt, 2009).

Alternative methods have attempted to mitigate these shortcomings. In particular, Imai and Ratkovic's (2014; 2015) covariate balancing propensity score (CBPS) method proposes a set of balancing conditions when estimating the propensity scores. Because it seeks to maximize the covariate balance between the treatment and control groups, this method is less sensitive to model misspecification than IPW. Fong et al. (2018) expand CBPS to accommodate continuous exposures, but the challenges involved in modeling conditional densities persist. With this in mind, they have also developed a

nonparametric extension that constructs weights that maximize the empirical likelihood while meeting a set of balancing conditions. Though the nonparametric CBPS (npCBPS) circumvents the need for specifying a functional form for the propensity score, it does so at a cost: since the empirical likelihood is not generally convex, the optimization procedure is often slow and may fail to find a solution. The latter can happen, for example, when we have a large number of covariates. The authors advance a workaround that adds flexibility to the covariate balancing conditions and penalizes the remaining imbalance. In doing so, they ensure that a weighting solution exists. Users can implement CBPS in R with the **CBPS** package (Fong et al., 2021).

Recently, Zhou and Wodtke (2020) propose the method of residual balancing weights (RBW) for estimating MSMs. RBW involves fitting models for the conditional means of post-treatment confounders given past treatments and confounders and extracting their residual terms. It then uses Hainmueller's (2012) entropy balancing method to find weights such that, in the weighted sample, 1) the residuals are orthogonal to future exposures, past treatments, and past confounders, and 2) the relative entropy between the weights and a set of base weights (e.g., survey sampling weights) is minimized. RBW is similar to npCBPS in that it relies on a set of balancing conditions to find the weights and does not require modeling the conditional distributions of the exposures. Both methods can, therefore, be easily adapted to cases where exposures are continuous.[1] Despite their similarities, RBW has a significant computational advantage: the relative entropy metric it uses to construct the weights leads to a convex optimization problem, so finding the weighting solution is computationally expeditious. As shown below, RBW manages to locate the solution considerably faster than npCBPS when we compare the methods' performance for the same problem.

In the sections that follow, we present an overview of the residual balancing method and how, in addition to contexts involving time-varying treatments, we can use it in cases of point treatments and causal mediation analysis. We then discuss the package that implements the method in R (**rbw**). Next, we describe the functions included in **rbw** and illustrate their use with various real-world data sets. The final section concludes.

## Overview of residual balancing

This section gives an overview of RBW. We first describe the notation used throughout the paper and briefly review MSMs. Next, we explain the underlying logic of RBW and provide an intuition for how the method works using a directed acyclic graph (DAG).

### Notation

Assume we have a study with $T \geq 2$ time points, and we are interested in the effect of a time-varying treatment, $A_t$ ($1 \leq t \leq T$), on some end-of-study outcome $Y$. We also have a vector of observed time-varying confounders, $L_t$, at each time point, which may be affected by prior treatments. $\bar{A}_t = (A_1, ..., A_t)$ and $\bar{L}_t = (L_1, ..., L_t)$ denote treatment and covariate histories up to time $t$. Furthermore, $\bar{A} = \bar{A}_T$ and $\bar{L} = \bar{L}_T$ represent a respondent's complete treatment and covariate histories, respectively. Finally, let $Y(\bar{a})$ be the potential outcome under some treatment history $\bar{a}$.

### MSMs

An MSM is a model for the marginal mean of the potential outcomes under some treatment history:

$$\mathbb{E}[Y(\bar{a})] = \mu(\bar{a}; \beta), \tag{1}$$

where $\mu(.)$ is some function and $\beta$ are a set of parameters capturing the causal effects of interest. We can identify an MSM from observed data under three assumptions:

1. consistency: if $\bar{A} = \bar{a}$, then $Y = Y(\bar{a})$;
2. sequential ignorability: at each time point $t$, treatment is unconfounded conditional on past treatments and the covariate history up to that point. Formally, $Y(\bar{a}) \perp\!\!\!\perp A_t | \bar{A}_{t-1}, \bar{L}_t$; and

[1] Other methods for estimating causal effects of continuous exposures include doubly robust estimators, which model both the treatment and outcome processes and give consistent estimates as long as one of these models is correctly specified. For example, Kennedy et al. (2017) introduce an approach that does not require any parametric assumptions about the effect curve. Instead, it uses flexible data-adaptive methods both to estimate the treatment and outcome models and to subsequently fit the dose-response curve. Readers can install the R package that implements this method from GitHub (Kennedy, 2021).

3. positivity: at each time point $t$, treatment assignment must not be deterministic. That is, if $f(\bar{A}_{t-1} = \bar{a}_{t-1}, \bar{L}_t = \bar{l}_t) > 0$, then $f(A_t = a_t | \bar{A}_{t-1} = \bar{a}_{t-1}, \bar{L}_t = \bar{l}_t) > 0$, where $f(\cdot)$ represents a probability mass or density function.

Under these assumptions, Robins (1986) shows that the expected value of the potential outcome $\mathbb{E}[Y(\bar{a})]$ can be identified via the g-computation formula:

$$\mathbb{E}[Y(\bar{a})] = \int ... \int \mathbb{E}[Y|\bar{A} = \bar{a}, \bar{L} = \bar{l}] \prod_{t=1}^{T} f(l_t|\bar{l}_{t-1}, \bar{a}_{t-1}) d\mu(l_t), \tag{2}$$

where $\mu()$ is an appropriate dominating measure. While Equation 2 provides a general formula for identifying causal effects in the presence of time-varying treatments, directly evaluating it is often impractical, particularly when we have many covariates and time periods.

### The rationale behind residual balancing

Now consider the formula for the conditional mean of the observed outcome $Y$ given some treatment history:

$$\mathbb{E}[Y|\bar{A} = \bar{a}] = \int ... \int \mathbb{E}[Y|\bar{A} = \bar{a}, \bar{L} = \bar{l}] \prod_{t=1}^{T} f(l_t|\bar{l}_{t-1}, \bar{a}) d\mu(l_t). \tag{3}$$

By comparing Equations 2 and 3, we see that weighting the observed population by

$$W_l = \prod_{t=1}^{T} \frac{f(L_t|\bar{L}_{t-1}, \bar{A}_{t-1})}{f(L_t|\bar{L}_{t-1}, \bar{A})} \tag{4}$$

creates a pseudo-population in which $f^*(l_t|\bar{l}_{t-1}, \bar{a}) = f^*(l_t|\bar{l}_{t-1}, \bar{a}_{t-1}) = f(l_t|\bar{l}_{t-1}, \bar{a}_{t-1})$ and $\mathbb{E}^*[Y|\bar{A} = \bar{a}] = \mathbb{E}^*[Y(\bar{a})] = \mathbb{E}[Y(\bar{a})]$, where * represents quantities in the pseudo-population. Estimating the conditional densities of Equation 4 is challenging because $L_t$ is often high-dimensional.

Zhou and Wodtke (2020) demonstrate that the condition $f^*(l_t|\bar{l}_{t-1}, \bar{a}) = f^*(l_t|\bar{l}_{t-1}, \bar{a}_{t-1}) = f(l_t|\bar{l}_{t-1}, \bar{a}_{t-1})$ implies a series of moment conditions in the pseudo-population. Most importantly,

$$\mathbb{E}^*[\delta(g(L_t))h(\bar{L}_{t-1}, \bar{A})] = \mathbb{E}^*[\delta(g(L_t))]\mathbb{E}^*[h(\bar{L}_{t-1}, \bar{A})] = 0, \tag{5}$$

where:

- $g(.)$ and $h(.)$ are scalar functions.
- $\delta(g(L_t))$ is the residual of $g(L_t)$ with respect to its conditional mean given the observed past: $\delta(g(L_t)) = g(L_t) - \mathbb{E}[g(L_t)|\bar{L}_{t-1}, \bar{A}_{t-1}]$.

Residual balancing aims to emulate the moment conditions (5) that would hold in the pseudo-population if it were possible to weight the observed population by $W_l$. To do so, the method 1) specifies a set of $g(\cdot)$ functions, $G(L_t) = \{g_1(L_t), ..., g_{J_t}(L_t)\}$ and a set of $h(.)$ functions, $H(\bar{L}_{t-1}, \bar{A}) = \{h_1(\bar{L}_{t-1}, \bar{A}), ..., h_{K_t}(\bar{L}_{t-1}, \bar{A})\}$; 2) computes a set of residual terms $\delta(g(L_t)) = g(L_t) - \mathbb{E}[g(L_t)|\bar{L}_{t-1}, \bar{A}_{t-1}]$; and 3) finds a set of weights such that, for any $j$, $k$, and $t$, the cross-moment of $\delta(g_j(l_{it}))$ and $h_k(\bar{l}_{i,t-1}, \bar{a}_i)$ is zero in the weighted data. That is, RBW locates the $rbw_i$ weights subject to the following balancing conditions:

$$\sum_{i=1}^{n} rbw_i c_{ir} = 0, \;\; 1 \leq r \leq n_c, \tag{6}$$

where $c_{ir}$ is the rth element of $\boldsymbol{c}_i = \{\delta(g_j(l_{it}))h_k(\bar{l}_{i,t-1}, \bar{a}_i); 1 \leq j \leq J_t, 1 \leq k \leq K_t, 1 \leq t \leq T\}$ and $n_c = \sum_{t=1}^{T} J_t K_t$ denotes the total number of balancing conditions. The residualized confounders at each time point are balanced across future treatments as well as past treatments and confounders (the observed past). RBW thus adjusts for post-treatment confounding without inducing overcontrol and collider-stratification bias.

Moreover, Zhou and Wodtke (2020) follow Hainmueller (2012) and minimize the relative entropy between $rbw_i$ and a set of base weights $q_i$ (e.g., survey sampling weights):

$$\min_{rbw_i} \sum_{i} rbw_i \log(rbw_i / q_i) \tag{7}$$

We can then use the method of Lagrange multipliers to find a weighting solution that minimizes the relative entropy between $rbw_i$ and $q_i$ subject to the $n_c$ balancing constraints. We discuss this procedure in greater depth below when describing the function `rbw::eb2()`. The convexity of the relative entropy metric renders it considerably more computationally efficient than nonconvex loss functions that can also be used to construct weights, such as the empirical likelihood (Fong et al., 2018).

A typical implementation of residual balancing can be summarized in three steps:

1. For each covariate $j$ and at each time point $t$, fit a linear, logistic, or Poisson regression of $l_{ijt}$ (depending on its level of measurement) on $\bar{l}_{i,t-1}$ and $\bar{a}_{i,t-1}$. Then compute the residuals $\hat{\delta}(l_{ijt})$. For the covariates in $L_1$ (the first time period), the residuals are computed as deviations from the sample mean: $\hat{\delta}(l_{ij1}) = l_{ij1} - \text{avg}(l_{j1})$. This step relies on the idea that $g_j(L_t) = L_{jt}$, where $L_{jt}$ is the jth element of the covariate vector $L_t$, is a natural choice for the set of $g(.)$ functions constituting $G(L_t)$.
2. Find a set of weights, $rbw_i$, such that:
   a) in the weighted sample, the residuals $\hat{\delta}(l_{ijt})$ are orthogonal to all future treatments and the regressors of $l_{ijt}$ (i.e., the past treatments and past confounders);
   b) the relative entroy between $rbw_i$ and the base weights $q_i$ is minimized.
3. Use the weights to fit an MSM.

Figure 1 depicts the logic of RBW in a DAG. Following the notation described above, $A_t$ denotes our time-varying treatment, $L_t$ is a vector of time-varying confounders, and $Y$ is our end-of-study outcome. Further, assume two time points, $t = 1, 2$. The $rbw_i$ weights break the dependence between $A_t$ and $\bar{L}_t$ at each time point. That is, the weights create a pseudo-population where the confounding arrows $L_1 \rightarrow A_1$, $L_1 \rightarrow A_2$, and $L_2 \rightarrow A_2$ are broken while all others are preserved. It is important to note that $L_1$ is *marginally* independent of both $A_1$ and $A_2$ in the pseudo-population, but $L_2$ is *conditionally* independent of $A_2$, given $L_1$ and $A_1$. Hence, RBW invokes a model for the conditional mean of $L_2$ given $L_1$ and $A_1$ and balances the residuals from this model across levels of $A_2$ and levels of $(L_1, A_1)$ (the observed past). This procedure avoids overcontrol and collider-stratification bias when adjusting for post-treatment confounding because it breaks the confounding arrow $L_2 \rightarrow A_2$ while leaving the causal arrow $A_1 \rightarrow L_2$ intact.

Finally, since $\mathbb{E}^*[Y|\bar{A} = \bar{a}] = \mathbb{E}^*[Y(\bar{a})] = \mathbb{E}[Y(\bar{a})]$ in the pseudo-population, we can estimate the marginal effects of interest by fitting a model for the conditional mean of the observed outcome given the treatment history (and possibly a set of baseline confounders) with weights equal to $rbw_i$.

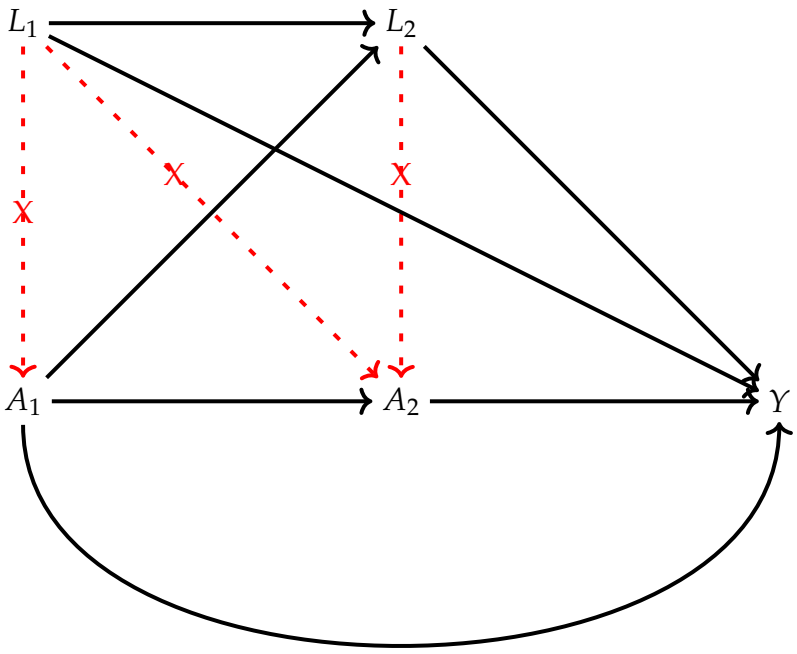


**Figure 1:** The underlying logic of residual balancing.

## Uses of residual balancing

The rationale described in the previous section is targeted at estimating the causal effects of time-varying treatments. With minor adaptations, we can expand the use of RBW to two other contexts commonly encountered in the social and biomedical sciences: point treatment situations and causal mediation analysis.

### RBW for estimating the average effect of a point treatment

RBW can be easily adapted to a point treatment situation where the user aims only to adjust for a set of baseline (i.e., time-invariant) confounders to estimate the average treatment effect. To this end, we modify the procedure above as follows:

1. Compute the response residuals $\hat{\delta}(x_{ij})$ for each baseline confounder $X_j$ by centering it around its sample mean: $\hat{\delta}(x_{ij}) = x_{ij} - \text{avg}(x_j)$.
2. Find a set of weights, $rbw_i$, such that:
    a) in the weighted sample, the residuals $\hat{\delta}(x_{ij})$ are orthogonal to the treatment;
    b) the relative entropy between $rbw_i$ and the base weights $q_i$ is minimized.
3. Use the weights to fit an MSM.

The DAG of Figure 2 illustrates the point treatment situation. $A$ represents the one-shot treatment, $X$ is a vector of baseline confounders, and $Y$ denotes our outcome. Weighting the observed population by $rbw_i$ mimics a pseudo-population where the link between $A$ and $X$ is broken. We can then fit a model for the conditional mean of $Y$ given $A$ to estimate the causal effects of interest.[2]

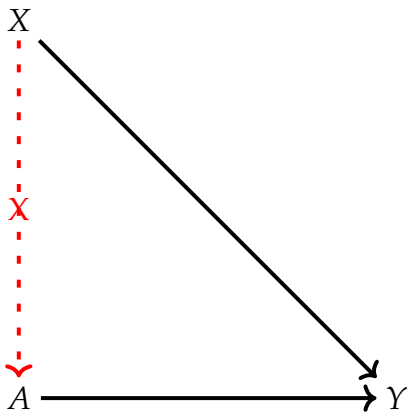


**Figure 2:** RBW in a Point Treatment.

### RBW in causal mediation analysis

In causal mediation analysis, researchers are often concerned with the joint effects of a one-shot treatment, $A$, and a mediator, $M$, on some end-of-study outcome $Y$ when both baseline confounders ($X$) and some post-treatment confounders for the mediator-outcome relationship ($Z$) are present. With minor adjustments, the RBW implementation for causal mediation analysis is similar to the case with time-varying treatments:

1. As in the point treatment scenario, compute the response residuals $\hat{\delta}(x_{ij})$ for each baseline confounder $X_j$ by centering it around its sample mean.
    - Note: users may skip this step by including the baseline confounders in the MSM in the final step.
2. Estimate the response residuals $\hat{\delta}(z_{ij})$ for each post-treatment confounder $Z_j$ by fitting a linear, logistic, or Poisson regression of $z_{ij}$ (depending on its level of measurement) on the treatment $a_i$ and the baseline confounders $x_i$: $\hat{\delta}(z_{ij}) = z_{ij} - \mathbb{E}[z_{ij}|a_i, x_i]$.

[2]The vector $X$ captures all confounding under the ignorability assumption.

3. Find a set of weights, $rbw_i$, such that:
    a) in the weighted sample, i) the baseline residuals $\hat{\delta}(x_{ij})$ are orthogonal to the treatment $a_i$ and the mediator $m_i$; and ii) the post-treatment residuals $\hat{\delta}(z_{ij})$ are balanced across the treatment $a_i$, the mediator $m_i$, and the baseline confounders $x_i$;
    b) the relative entropy between $rbw_i$ and the base weights $q_i$ is minimized.
4. Use the weights to fit an MSM for the joint effects of the treatment and the mediator on the outcome:
    a) In causal mediation analysis, the potential outcomes of interest are denoted by $Y(a, m)$ (this is the potential outcome under treatment $a$ and mediator value $m$). We can then express a saturated MSM as $\mathbb{E}[Y(a, m)] = \alpha_0 + \alpha_1 a + \alpha_2 m + \alpha_3 am$.
    b) Alternatively, the baseline confounders can be included in the MSM if users decide to skip the first step: $\mathbb{E}[Y(a, m)|X] = \alpha_0 + \alpha_1 a + \alpha_2 m + \alpha_3 am + \alpha_4^T X$.
    c) Finally, the controlled direct effects (CDE) of the treatment can be estimated as $\widehat{\text{CDE}}(m) = \mathbb{E}[Y(1, m) - Y(0, m)] = \hat{\alpha}_1 + \hat{\alpha}_3 m$. The CDE measures the causal effect of the treatment on the outcome when the mediator is fixed at some value $m$ for all units.

## The R package

The R package **rbw** contains four functions:

- `eb2()`, for generating minimum entropy weights subject to a set of balancing constraints.
- `rbwPoint()`, for constructing residual balancing weights to estimate the causal effects of one-shot treatments.
- `rbwMed()`, for constructing residual balancing weights to estimate controlled direct effects in causal mediation analysis.
- `rbwPanel()`, for constructing residual balancing weights to estimate the marginal effects of time-varying treatments.

Next, we explain each of these functions. The package also includes several real-world data sets (`advertisement`, `peace`, `campaign_long`, and `campaign_wide`), which we describe and analyze in the examples below.

### Function `eb2()`

This function is an adaptation of `ebal::eb()` (Hainmueller, 2014). It is called internally by other functions in **rbw** to implement the method of Lagrange multipliers for locating a weighting solution that minimizes the relative entropy between $rbw_i$ and $q_i$ subject to the set of $n_c$ balancing constraints described in Equation 6. Zhou and Wodtke (2020) impose an additional normalization constraint that ensures that the residual balancing weights sum to the sample size: $\sum_i rbw_i = n$. Following Hainmueller (2012), the authors obtain the primal optimization problem:

$$\min_{rbw_i} L^p = \sum_{i=1}^{n} rbw_i \log(rbw_i/q_i) + \sum_{r=1}^{n_c} \lambda_r \sum_{i=1}^{n} rbw_i c_{ir} + \lambda_0 \left( \sum_{i=1}^{n} rbw_i - n \right), \tag{8}$$

where $\{\lambda_1, ..., \lambda_{n_c}\}$ are the Lagrange multipliers for the balancing constraints and $\lambda_0$ is the Lagrange multiplier for the normalization constraint. Since the loss function $L^p$ is strictly convex, the first order condition of $\frac{\partial L^p}{\partial rbw_i} = 0$ implies that the solution for each weight is

$$rbw_i^* = \frac{nq_i \exp(-\sum_{r=1}^{n_c} \lambda_r c_{ir})}{\sum_{i=1}^{n} q_i \exp(-\sum_{r=1}^{n_c} \lambda_r c_{ir})}. \tag{9}$$

We can then insert Equation 9 into $L^p$, leading to an unrestricted dual problem:

$$\max_{\lambda_r} L^d = -\log \left( \sum_{i=1}^{n} q_i \exp \left( -\sum_{r=1}^{n_c} \lambda_r c_{ir} \right) \right), \tag{10}$$

or equivalently,

$$\min_{Z} L^d = \log \left( Q' \exp(CZ) \right), \tag{11}$$

where $Q = [q_1, ..., q_n]'$, $C = [c_1, ..., c_1]'$, and $Z = -[\lambda_1, ..., \lambda_{n_c}]'$. Since $L^d$ is strictly convex, the solution is guaranteed to be unique — assuming one exists. Given that both the gradient and the Hessian have closed-form expressions, we can solve the problem using Newton's method. `eb2()` implements the algorithm. If convergence is successful, the function tells the user that "Entropy minimization converged within tolerance level." Otherwise, it warns that entropy minimization did not converge and suggests increasing the number of iterations or reducing the number of balancing constraints.

The convexity of our optimization problem leads to appreciable computational gains over other methods that use alternative loss functions. This will be demonstrated later when we compare the performance of RBW with that of npCBPS — which uses the empirical likelihood — for the same problem.

`eb2()` is used as:

```
eb2(C, M, Q, Z = rep(0, ncol(C)), max_iter = 200, tol = 1e-04, print_level = 1)
```

and takes the following arguments:

- `C` is a constraint matrix, with each column corresponding to a balancing constraint.
- `M` is a vector of moment conditions to be met in the reweighted sample (per Equation 6, this is a vector of zeros with length equal to the number of columns of `C` when the other functions in `rbw` call `eb2()` internally).
- `Q` is a vector of base weights.
- `Z` is a vector of Lagrange multipliers to be initialized.
- `max_iter` determines the maximum number of iterations for Newton's method.
- `tol` is a tolerance parameter used to determine convergence. Specifically, convergence is achieved if `tol` is greater than the maximum absolute value of the deviations between the moments of the reweighted data and the target moments (i.e., `M`).
- `print_level` determines the level of printing:
  - **1** normal: print whether the algorithm converges or not.
  - **2** detailed: print also the maximum absolute value of the deviations between the moments of the reweighted data and the target moments in each iteration.
  - **3** very detailed: print also the step length of the line searcher in iterations where a full Newton step is excessive

The output returned by `eb2()` is a list containing the following elements:

- `W` is a vector of normalized minimum entropy weights.
- `Z` is a vector of Lagrange multipliers.
- `converged` is a logical indicator for convergence.
- `maxdiff` is a scalar indicating the maximum absolute value of the deviation between the moments of the reweighted data and the target moments in each iteration.

### Function `rbwPoint()`

This function produces residual balancing weights to be used in a point treatment situation. It first takes a set of baseline confounders and computes the residuals for each confounder by centering it around its sample mean. Then it calls `eb2()` to find a set of weights, $rbw_i$, such that 1) the baseline residuals are orthogonal to the treatment in the weighted sample, and 2) the relative entropy between $rbw_i$ and the base weights is minimized. Additionally, `rbwPoint()` calls a function that ensures that the matrix of balancing constraints comprises only linearly independent columns.

`rbwPoint()` is used as:

```
rbwPoint(treatment, data, baseline_x, base_weights, max_iter = 200,
    tol = 1e-04, print_level = 1)
```

and takes the following arguments:

- `max_iter`, `print_level`, and `tol` have the same definitions as in `eb2()`.
- `treatment` is a symbol or character string for the treatment variable.
- `data` is a data frame containing all variables in the model.
- `baseline_x` is an expression for a set of baseline confounders stored in `data` or a character vector of the names of these variables.
- `base_weights` is an *optional* vector of base weights (or its name). If no value is supplied, the function sets a vectors of ones with length equal to the sample size as the base weights.

The output returned by `rbwPoint()` is a list containing the following elements:

- `weights` is a vector of residual balancing weights.
- `constraints` is a matrix of linearly independent residual balancing constraints.
- `eb_out` contains the results from calling the `eb2()` function.
- `call` is the matched call (the function call with all arguments specified by their full names).

### Function `rbwMed()`

This function produces residual balancing weights for causal mediation analysis. It takes an *optional* set of baseline confounders (as explained above, users can opt instead to include these covariates in the MSM later) and a list of model objects for the conditional mean of each post-treatment confounder given the treatment and baseline confounders. It then calls `eb2()` to find a set of weights, $rbw_i$, such that, in the weighted sample, 1) the baseline residuals are orthogonal to the treatment and the mediator; 2) the post-treatment confounders are balanced across the treatment, the mediator, and the baseline confounders; and 3) the relative entropy between $rbw_i$ and the base weights is minimized.

`rbwMed()` takes an additional argument, `interact`, a logical variable indicating whether the baseline and post-treatment confounders should be balanced against the treatment-mediator interaction. This argument is set to `FALSE` by default, but users suspecting an interaction effect may find it prudent to balance against it. Like `rbwPoint()`, `rbwMed()` calls a function internally to ensure that only linearly independent columns constitute the matrix of balancing constraints.

It is used as:

```
rbwMed(treatment, mediator, zmodels, data, baseline_x, interact = FALSE,
    base_weights, max_iter = 200, tol = 1e-04, print_level = 1)
```

and takes the following arguments:

- `treatment`, `data`, `base_weights`, `max_iter`, `print_level`, and `tol` are defined as in `rbwPoint()`.
- `baseline_x` is an *optional* expression for a set of baseline confounders stored in `data` or a character vector of the names of these variables.
- `mediator` is a symbol or character string representing the mediator variable.
- `zmodels` is a list of fited `lm` or `glm` objects for post-treatment confounders of the mediator-outcome relationship. Users should set this argument to `NULL` if there are no post-treatment confounders.
- `interact` is a logical variable indicating whether baseline and post-treatment confounders should be balanced against the treatment-mediator interaction term(s).

The output returned by `rbwMed()` is a list containing the same elements as the output from `rbwPoint()`.

### Function `rbwPanel()`

This function produces residual balancing weights for estimating the marginal effects of time-varying treatments. It takes a list of model objects for the conditional mean of each post-treatment confounder given past treatments and past confounders. Then it calls `eb2()` to find a set of weights, $rbw_i$, such that 1) residuals of the post-treatment confounders are orthogonal to future treatments and the observed past in the weighted sample, and 2) the relative entropy between $rbw_i$ and the base weights is minimized. Like the other functions, `rbwPanel()` ensures that the matrix of balancing constraints consists only of linearly independent columns.

It is used as:

```
rbwPanel(treatment, xmodels, id, time, data, base_weights, future = 1L,
    max_iter = 200, tol = 1e-04, print_level = 1)
```

and takes the following arguments:

- `data`, `base_weights`, `max_iter`, `print_level`, and `tol` are defined as in `rbwPoint()` and `rbwMed()`.
- `treatment` is a symbol or character string for the time-varying treatment.
- `xmodels` is a list of fitted `lm` or `glm` objects for time-varying confounders.
- `id` is a symbol or character string for the unit id variable.
- `time` is a symbol or character string for the time variable.
- `future` is an integer indicating the number of future treatments in the balancing conditions. When `future >0`, the residualized time-varying covariates are balanced not only with respect to current treatment $A_t$, but also with respect to future treatments $A_{t+1}$, ..., $A_{t+\texttt{future}}$. The default, `future = 1`, assumes away higher-ordered lagged effects of the covariates on the treatment. Users can leave out lagged effects entirely by setting `future` to zero.

The output returned by `rbwPanel()` is essentially the same as those from `rbwPoint()` and `rbwMed()`. The only difference is that the `weights` object, instead of a vector, is now a data frame with two columns: the id variable and the residual balancing weights (the column storing the weights is called `rbw`).

## Examples

We now illustrate the functions `rbwPanel()`, `rbwPoint()`, and `rbwMed()` with data sets `advertisement`, `peace`, `campaign_long`, and `campaign_wide`, which are included in **rbw**. We expect users to rarely need to call `eb2()` manually since its primary use is to be called internally by the other functions. Hence, we see little gain in providing a separate example for it.

### Point treatment: effects of political advertisements on campaign contributions

Urban and Niebler (2014) studied the effects of televised political advertisements on campaign contributions. Presidential candidates do not tend to deliberately advertise in states where competition for electoral votes is tame. Yet, some areas of noncompetitive states have overlapping media markets with battleground states. Because these media market spillovers do not encompass other forms of campaigning (e.g., rallies, speeches, etc.), the authors can isolate the effect of television advertising by restricting their analyses to noncompetitive states. Their original method involved estimating the propensity score with a logistic model and then conducting propensity score matching. To do so, they dichotomized the political advertising variable to indicate whether zip codes received more than 1000 advertisements.

Deeming this approach inadequate — in part because balancing against a dichotomous treatment does not ensure covariate balance on the underlying continuous variable — Fong et al. (2018) revisit the study using the CBPS method applied to a continuous treatment. Next, we examine how RBW fares compared with CBPS and IPW in this point treatment situation.

Importantly, CBPS assumes that the treatment variable is normally distributed. To satisfy this assumption, Fong et al. (2018) search across Box-Cox transformations to find the most appropriate transformation of the treatment. Instead, we rely on a simple log transformation in our current example. Q-Q plots show no sizable differences between the two approaches in achieving normality, so we favor the simpler alternative because it avoids extraneous details that divert the example from its primary objective — that is, illustrating RBW's applicability to a point treatment scenario. Hence, we have $a_i^* = \log(a_i + 1)$, where $a_i$ is the original treatment variable, the total number of political advertisements in a zip code, and $a_i^*$ is the transformed treatment. We also have a vector of baseline confounders consisting of the zip code's log population, population density, log median income, percent Hispanic, percent black, percent over age 65, percent college graduates, and a binary indicator of whether it is possible to commute to the zip code from a competitive state. Finally, $Y$ represents the outcome, campaign contributions in thousands of dollars.

Our MSM includes the transformed treatment variable and state dummies $U$ to account for state fixed effects:

$$\mathbb{E}[Y(a^*)|U] = \theta_0 + \theta_1 a^* + \theta_2^T U. \tag{12}$$

The data set `advertisement` contains the variables necessary for the analyses. We start by loading the necessary packages and our data set:

```
library("rbw")
data("advertisement")
```

`rbw::rbwPoint()` will construct the residual balancing weights by following the steps described above. It first computes the baseline residuals $\hat{\delta}(x_{ij}) = x_{ij} - \text{avg}(x_j)$ and then finds a set of weights such that, in the weighted sample, $\hat{\delta}(x_{ij})$ are orthogonal to the treatment, and the relative entropy between the residual balancing weights and a set of base weights is minimized. Since `advertisement` does not include sampling weights, `rbw::rbwPoint()` will set a vector of ones with length equal to the sample size as the base weights.

```
rbwPoint_fit <-
  rbwPoint(
    treat,
    baseline_x = c(
      log_TotalPop,
      PercentOver65,
```

```
      log_Inc,
      PercentHispanic,
      PercentBlack,
      density,
      per_collegegrads,
      CanCommute
    ),
    data =
      advertisement
  )
```

Next, we attach the $rbw_i$ weights to the data:

```
advertisement$rbwPoint_weights <- rbwPoint_fit$weights
```

Following most applications of MSMs, we compute standard errors using the robust ("sandwich") variance estimator. This can be implemented with the function `survey::svydesign()`, which allows us to specify a complex survey design and estimate standard errors consistent with this specification.[3]

```
library("survey")
rbwPoint_design <- svydesign(ids = ~ 1,
                             weights = ~ rbwPoint_weights,
                             data = advertisement)
```

We then use the residual balancing weights to fit the MSM defined in Equation 12:

```
rbwPoint_msm <- svyglm(Cont ~ treat + factor(StFIPS),
                       design = rbwPoint_design)
```

Since we have transformed our treatment variable, we need to make the necessary adjustments to compute the treatment effect. As Urban and Niebler (2014) suggest, it is informative to study the effect of going from zero to 1,000 political advertisements on campaign contributions. Hence, we create a `dose` variable to account for this. If $a_i^* = \log(a_i + 1)$, we can define our dose as $\log(1000 + 1)$:

```
dose <- log(1000 + 1)
```

To find the estimate $\hat{\tau}_{rbw}$ of the average treatment effect, we multiply the coefficient for the transformed treatment variable by our dose. We also multiply it by 1,000 since the outcome variable is measured in thousands of dollars:

```
rbwPoint_tau <- 1000 * coef(rbwPoint_msm)[2] * dose
```

Next, we use the basic properties of the variance operator to derive the standard error. Again, we multiply the result by 1,000 given the scale of the outcome variable:

```
rbwPoint_vcov <- stats::vcov(rbwPoint_msm)
rbwPoint_se <- 1000 * dose * sqrt(rbwPoint_vcov[2, 2])
```

We also report the results using the functions from the **CBPS** and **ipw** packages. Recall from the introduction that the parametric CBPS is similar to IPW in that it requires explicit models for the conditional distributions of the treatment. However, it improves IPW by considering a set of balancing conditions during the propensity score estimation, thereby reducing sensitivity to model misspecification. By contrast, the nonparametric CBPS (npCBPS) does not require direct estimation of the propensity score; instead, it finds weights that maximize the empirical likelihood while meeting a set of balancing constraints. As such, like RBW, it avoids the need to specify a propensity score model. Readers can find the code detailing the construction of the CBPS and IPW weights in the supplementary material.

Table 1 summarizes the findings. All methods yield relatively similar point estimates and standard errors. In particular, they indicate that going from zero to 1,000 political advertisements increases campaign contributions by around four thousand dollars, on average, although the point estimates from CBPS and IPW are slightly larger than those produced by RBW and npCBPS. The last column shows that different loss functions for the optimization problem can lead to stark differences in computation time. While RBW's relative entropy metric leads to a convex optimization problem that Newton's method can solve in less than one second, npCBPS's algorithm takes much longer to run.

[3] Zhou and Wodtke (2020) conduct various simulation studies to assess the performance of the robust variance estimator across different methods. They find that the estimator tends to overestimate the true sampling variance for residual balancing across nearly all scenarios, making it consistently conservative for RBW. By contrast, the estimator sometimes overestimates and other times underestimates the true sampling variance for other weighting methods, including CBPS.

**Table 1:** Comparison of RBW, npCBPS, CBPS, and IPW for a Point Treatment Situation

| Method | Estimate | Standard Error | Computation Time (in Seconds) |
|---|---|---|---|
| RBW | 4043 | 2131 | 0.26 |
| npCBPS | 3916 | 2091 | 39.23 |
| CBPS | 4181 | 2118 | 4.39 |
| IPW | 4118 | 2078 | 0.02 |

Computation time may differ depending on system setup.
System setup used to generate the results:
MacBook Pro (15-inch, 2018), 2.2 GHz 6-Core Intel Core i7, 16GB RAM.

We next illustrate the use of `rbwMed()` and `rbwPanel()`.

### Causal mediation analysis: the controlled direct effect of shared democracy on public support for war

A stylized fact in political science is that democracies do not engage in war with one another. To assess the role of public opinion in keeping the peace, Tomz and Weeks (2013) designed survey experiments that presented participants with a situation where a country was developing nuclear weapons. When describing the situation, the authors randomly and independently changed three characteristics of the country: its political regime (whether it was a democracy), alliance status (whether it had signed a military alliance with the United States), and economic ties (whether it had high levels of trade with the US). The outcome of interest was the respondent's support for military action on a five-point scale ranging from "oppose strongly" to "favor strongly." The authors found that respondents were considerably less supportive of military action against democracies than otherwise identical autocratic regimes.

They then went on to investigate the causal mechanisms through which shared democracy reduces public enthusiasm for war. In particular, Tomz and Weeks (2013) measured individuals' beliefs about the level of threat posed by the potential adversary, the cost of the intervention, and the likelihood of success. They also collected data on people's moral concerns about using military force. Their methodological framework focuses on estimating the natural direct and natural indirect effects (Imai et al., 2011, 2010), whose identification assumptions require that no post-treatment confounding of the mediator-outcome relationship exists. As such, the authors examined the role of each mediator separately by assuming they operate independently of one another. Yet, as discussed in Zhou and Wodtke (2020), beliefs about the threat, cost, and likelihood of success may affect an individual's perceptions of morality while also influencing support for war directly. By treating these variables as post-treatment confounders, Zhou and Wodtke (2020) analyzed the mediating role of morality using controlled direct effects.

Let $A$ denote the treatment, whether the country developing nuclear weapons was presented as a democracy, $M$ the mediator, a dummy variable indicating whether the participant deemed the military action morally wrong, and $Y$ the outcome, the respondent's support for war on a five-point scale. We also have a set of baseline confounders ($X$) including dummies for the other two randomized treatments (alliance status and economic ties) in addition to several demographic and attitudinal controls.[4] Finally, $Z$ is a vector of post-treatment confounders comprising measures of beliefs about the threat, cost, and likelihood of success.

Our MSM is thus defined as:

$$\mathbb{E}[Y(a,m)] = \alpha_0 + \alpha_1 a + \alpha_2 m + \alpha_3 am. \tag{13}$$

We can alternatively include the baseline confounders in the MSM instead of balancing them across the treatment and the mediator with baseline residuals:

$$\mathbb{E}[Y(a,m)|X] = \alpha_0 + \alpha_1 a + \alpha_2 m + \alpha_3 am + \alpha_4^T X. \tag{14}$$

The `peace` data set includes the variables we will use in the analyses. Let us first consider the approach where the baseline confounders are balanced across the treatment and the mediator using the baseline residuals computed from centering each covariate around its sample mean.

[4]For example, attitudinal controls include respondents' attitudes toward ethnocentrism measured with a series of questions about their opinions on immigration, affirmative action, and gay marriage.

As explained above, RBW in causal mediation analysis requires models for the conditional means of each post-treatment confounder $Z_j$ given the treatment $a_i$ and the baseline confounders $x_i$: $\hat{\mathbb{E}}[z_{ij}|a_i, x_i]$. Hence, we assume that `threatc`, `cost`, and `successc` are measured on a continuous scale and fit linear models for each:

```
data("peace")
z1 <- lm(threatc ~ ally + trade + h1 + i1 + p1 + e1 + r1 +
           male + white + age + ed4 + democ,
         data = peace)
z2 <- lm(cost ~ ally + trade + h1 + i1 + p1 + e1 + r1 +
           male + white + age + ed4 + democ,
         data = peace)
z3 <- lm(successc ~ ally + trade + h1 + i1 + p1 + e1 + r1 +
           male + white + age + ed4 + democ,
         data = peace)
```

We then store the three model objects together in a list to be passed later to the `zmodels` argument in `rbwMed()`:

```
zmodels <- list(z1, z2, z3)
```

To construct the residual balancing weights, `rbwMed()` will 1) compute the baseline residuals $\hat{\delta}(x_{ij}) = x_{ij} - \text{avg}(x_j)$ and the post-treatment residuals $\hat{\delta}(z_{ij}) = z_{ij} - \mathbb{E}[z_{ij}|a_i, x_i]$ and 2) find a set of weights such that a) in the weighted sample, the baseline and post-treatment residuals meet the orthogonality requirements described earlier, and b) the relative entropy between the residual balancing weights and a set of base weights is minimized. The function will use a vector of ones as the base weights since `peace` does not include sampling weights. We also pass the name of our mediator to the `mediator` argument:

```
rbwMed_fit <- rbwMed(
  treatment = democ,
  mediator = immoral,
  zmodels = zmodels,
  baseline_x = c(ally, trade, h1, i1,
                 p1, e1, r1, male, white, age, ed4),
  data = peace
)
```

Next, we attach the weights to `peace`:

```
peace$rbwMed_weights <- rbwMed_fit$weights
```

We use `survey::svydesign()` to estimate robust standard errors:

```
rbwMed_design <- svydesign(ids = ~ 1,
                           weights = ~ rbwMed_weights,
                           data = peace)
```

Finally, we use the residual balancing weights to fit the MSM from Equation 13:

```
rbwMed_msm <- svyglm(strike ~ democ * immoral,
                     design = rbwMed_design)
```

Steps are similar for the case where the baseline confounders are not balanced against the treatment and the mediator but instead adjusted in the MSM, as defined in Equation 14. The models for the conditional means of the post-treatment confounders are the same as before, but we now leave the `baseline_x` argument empty:

```
rbwMed2_fit <- rbwMed(
  treatment = democ,
  mediator = immoral,
  zmodels = zmodels,
  data = peace
)
peace$rbwMed2_weights <- rbwMed2_fit$weights
rbwMed2_design <- svydesign(ids = ~ 1,
                            weights = ~ rbwMed2_weights,
```

```
                              data = peace)
rbwMed2_msm <- svyglm(strike ~ ally + trade + h1 + i1 + p1 +
                        e1 + r1 + male + white + age + ed4 + democ * immoral,
                      design = rbwMed2_design)
```

We summarize the results in Table 2. The estimated CDE if respondents lacked any moral qualms about military intervention, i.e., $\widehat{\text{CDE}}(m = 0) = \hat{\mathbb{E}}[Y(1,0) - Y(0,0)] = \hat{\alpha}_1$, is $-0.32$ under the first approach and $-0.36$ under the second. Both estimates are statistically significant at the level of 0.01. The estimated CDE if respondents had moral qualms about military intervention, i.e., $\widehat{\text{CDE}}(m = 1) = \hat{\mathbb{E}}[Y(1,1) - Y(0,1)] = \hat{\alpha}_1 + \hat{\alpha}_3$, is $-0.36$ under the first approach and $-0.22$ under the second.

**Table 2:** MSM Results for the Controlled Direct Effects of Shared Democracy on Public Support for War

| | Baseline Confounders Balanced | Baseline Confounders Adjusted in the MSM |
|---|---|---|
| Shared democracy | $-0.317^{***}$ | $-0.360^{***}$ |
| | (0.098) | (0.080) |
| Moral concerns | $-1.272^{***}$ | $-1.201^{***}$ |
| | (0.175) | (0.135) |
| Shared democracy * Moral concerns | $-0.048$ | 0.138 |
| | (0.210) | (0.157) |

$^{*}$p<0.1; $^{**}$p<0.05; $^{***}$p<0.01
Robust standard erros in parentheses.

### Time-varying treatment: the cumulative effect of negative advertising on vote shares

We conclude this section with an example of RBW applied to a context involving time-varying treatments and confounders. Political scientists have shown interest in examining the cumulative effect of negative campaign advertising on election outcomes (Lau et al., 2007; Blackwell, 2013; Imai and Ratkovic, 2015). This is an intricate process because while polling results affect current campaign strategies, they are also constantly shifting, as they respond both to previous results and candidates' use of negative advertising in the past. For their ability to accommodate dynamic causal relationships — particularly since they allow past treatments to affect current outcomes (i.e., "carryover effects") and past outcomes to influence current treatment (i.e., "feedback effects") (Imai and Kim, 2019) — MSMs are suitable for this research question.

Let $A_t$ represent our continuous treatment, the proportion of campaign advertisements mentioning the adversary in each campaign week, $L_t$ our time-varying confounders, the Democratic share and the share of undecided voters in the polls, and $Y$ the outcome, the Democratic share of the two-party vote. Additionally, we have a set of baseline confounders $X$ including total campaign length, election year, and whether the election is senatorial or gubernatorial.

Zhou and Wodtke (2020) define an MSM as follows:

$$\mathbb{E}[Y(\bar{a})|X] = \beta_0 + \beta_1 \text{avg}(\bar{a}) + \beta_2 V + \beta_3 \text{avg}(\bar{a}) V + \beta_4^T X, \tag{15}$$

where $V$ is an indicator of incumbency status used to construct interactions that allow the effect to differ between incumbents and nonincumbents, and $\text{avg}(\bar{a})$ is the average proportion of advertisements that were negative over the final five weeks of the campaign multiplied by ten (we multiply by ten following Zhou and Wodtke (2020) so that regression coefficients can be interpreted as the effect of a ten-percentage point increase in negative advertising).

**rbw** contains two data sets associated with this problem: `campaign_long` and `campaign_wide`. They represent, respectively, the long-format and wide-format data on negative campaign advertising.

Recall that RBW requires us to fit a model for the conditional mean of each covariate at each time point given the observed past. We thus estimate regression models of our time-varying confounders $L_t$ ($t \geq 2$) on lagged treatment and lagged confounders. We also interact each regressor with the week dummies, thus allowing the coefficients to change over time in a flexible manner:

```
data("campaign_long")
data("campaign_wide")
x1 <-
  lm(dem.polls ~ (neg.dem.l1 + dem.polls.l1 + undother.l1) * factor(week),
     data = campaign_long)
x2 <-
  lm(undother ~ (neg.dem.l1 + dem.polls.l1 + undother.l1) * factor(week),
     data = campaign_long)
```

We then create a list with the model objects to be passed later to the `xmodels` argument in `rbwPanel()`:

```
xmodels <- list(x1, x2)
```

To construct the residual balancing weights, `rbwPanel()` first extracts the residual terms $\hat{\delta}(L_t)$ from the models above. Note that for each covariate in $L_1$ (the first time period), the residuals are computed as deviations from the sample mean. Then the function finds a set of weights, such that, in the weighted sample, the residuals are orthogonal to current and future treatments as well as the regressors of $L_{jt}$, and the relative entropy between the residual balancing weights and the base weights is minimized. Note that we set the `future` argument to zero to replicate the results from Zhou and Wodtke (2020) since the authors balance the residualized time-varying confounders only with respect to the current treatment, thus assuming away lagged effects of the covariates on the treatment.[5] Since our data do not include sampling weights, a vector of ones is used as the base weights. Additionally, we need to pass arguments indicating the unit id and the time variables due to the longitudinal structure of our data set.

```
rbwPanel_fit <- rbwPanel(
  treatment = neg.dem,
  xmodels = xmodels,
  id = id,
  time = week,
  data = campaign_long,
  future = 0
)
```

We now attach the weights to `campaign_wide` using the pipe operator and the `left_join()` function from the package **dplyr** (this merging is permitted because the `weights` object returned by `rbwPanel()` is a data frame containing the id variable and the residual balancing weights):

```
library("dplyr")
campaign_wide <- campaign_wide %>%
  left_join(rbwPanel_fit$weights, by = "id")
```

We use the functions from **dplyr** to rename the weights so that our variable's name is consistent with the previous examples:

```
campaign_wide <- campaign_wide %>%
  rename(rbwPanel_weights = rbw)
```

Again, we use `survey::svydesign()` to compute robust standard errors:

```
rbwPanel_design <- svydesign(ids = ~ 1,
                             weights = ~ rbwPanel_weights,
                             data = campaign_wide)
```

Finally, we use the residual balancing weights to fit the MSM from Equation 15:

```
rbwPanel_msm <- svyglm(demprcnt ~ ave_neg * deminc + camp.length +
                         factor(year) + office,
                       design = rbwPanel_design)
```

We report the model results in Table 3. The estimate for nonincumbents is 0.49 and for incumbents is $0.49 - 1.48 = -0.99$. The effect of negative advertisement for nonincumbents is positive though not statistically significant — a ten percentage point increase in the proportion of negative advertising throughout the last five weeks of the campaign increases the Democratic vote share by about half a percentage point, on average. However, the interaction term is significant at the level of 0.01, and

**Table 3:** MSM Results for the Cumulative Effect of Negative Advertising on Vote Shares

| | |
|---|---|
| Average proportion | 0.490 |
| | (0.315) |
| Incumbency | 14.971*** |
| | (2.823) |
| Average proportion * Incumbency | −1.484*** |
| | (0.531) |

*p<0.1; **p<0.05; ***p<0.01
Robust standard errors in parentheses.

incumbents see a sizeable negative effect from negative advertising: a ten percentage point increase in negative advertising reduces a candidate's vote share by about one percentage point, on average.

We conclude by comparing RBW's computational performance to CBPS for a dichotomized version of the treatment representing whether more than 10% of the candidate's advertising was negative for each week. We use this dichotomized variable because CBPS has not been extended to work with continuous treatments in longitudinal settings. Additionally, we do not compare RBW to npCBPS because the latter's use is restricted to point treatment situations. Lastly, though we can construct IPW weights almost immediately, Zhou and Wodtke (2020) show that IPW yields considerably larger effect estimates than RBW and CBPS for this particular case (likely due to the method's susceptibility to model misspecification), so we do not report the IPW results.

Let us first construct the RBW weights. The steps are identical to the ones detailed above for the continuous treatment, the only change being the name of our treatment variable. Next, we use the `CBMSM()` function from the `CBPS` package to generate our CBPS weights. To facilitate comparisons of computation time, we also call `Sys.time()` before and after running the functions that produce the weights in each package. The scalar objects `rbwPanel_time` and `CBPSPanel_time` store how long each method takes to construct the corresponding weights.

```
m1 <-
  lm(dem.polls ~ (d.gone.neg.l1 + dem.polls.l1 + undother.l1) * factor(week),
     data = campaign_long)
m2 <-
  lm(undother ~ (d.gone.neg.l1 + dem.polls.l1 + undother.l1) * factor(week),
     data = campaign_long)

xmodels <- list(m1, m2)

rbwPanel_start <- Sys.time()
rbwPanel_fit <- rbwPanel(
  treatment = neg.dem,
  xmodels = xmodels,
  id = id,
  time = week,
  data = campaign_long,
  future = 0
)
rbwPanel_end <- Sys.time()
rbwPanel_time <- rbwPanel_end - rbwPanel_start

CBPSPanel_form <-
  "d.gone.neg ~ d.gone.neg.l1 + dem.polls + undother + camp.length + deminc + office + factor(year)"

CBPSPanel_start <- Sys.time()
CBPSPanel_fit <-
  CBMSM(
    formula = CBPSPanel_form,
    time = campaign_long$week,
```

[5]The negative advertising data set spans five weeks, so users interested in considering lagged effects of the time-varying covariates on the treatment may set `future > 0`, up to `future = 4`.

```
    id = campaign_long$demName,
    data = campaign_long,
    type = "MSM",
    iterations = NULL,
    twostep = TRUE,
    msm.variance = "approx",
    time.vary = TRUE
  )
CBPSPanel_end <- Sys.time()
CBPSPanel_time <- CBPSPanel_end - CBPSPanel_start
```

The output below shows the computation times:

```
rbwPanel_time

#> Time difference of 0.04886699 secs

CBPSPanel_time

#> Time difference of 28.32747 secs
```

Whereas RBW takes less than one second to construct the weights, CBPS takes much longer. Hence, the longitudinal setting presents the same pattern we saw above for the point treatment situation: RBW has considerable gains in computational performance over alternative methods of constructing weights for MSMs. Since our focus has been on comparing computational performances, we omit the effect estimates for the dichotomized treatment. As shown in Zhou and Wodtke (2020), the results are broadly consistent with those based on the continuous treatment, with RBW and CBPS yielding similar point estimates.

## Conclusion

Compared to other methods of constructing weights for MSMs, RBW has several advantages. In particular, it does not require modeling the conditional distributions of exposures and is thus easy to use with continuous treatments. Previous simulation studies suggest that it is often more efficient and more robust to model misspecification than alternative weighting strategies (Zhou and Wodtke, 2020). RBW is also favorable from a computational perspective. Its procedure for finding weights involves a convex optimization problem, allowing RBW to locate a solution substantially faster than alternative methods whose optimization relies on nonconvex loss functions — such as the recently proposed nonparametric version of CBPS, which uses the empirical likelihood (Fong et al., 2018). Table 4 sums up these comparisons.

**Table 4:** Comparison of Methods and Software Implementation

| Method | Models for Conditional Distributions of Exposures | Balancing Constraints | Implemented for Time-varying Treatments? | R Package |
|---|---|---|---|---|
| IPW | Required | Absent | Yes | **ipw** |
| CBPS | Required | Present | Yes | **CBPS** |
| npCBPS | Not Required | Present | No | **CBPS** |
| RBW | Not Required | Present | Yes | **rbw** |

After explaining the underlying logic of RBW, we have described its implementation in the R package **rbw**. With examples from several data sets, we have demonstrated the use of **rbw** in three distinct contexts: effect estimation for point treatments, causal mediation analysis, and effect estimation for time-varying treatments with time-varying confounders.

Nonetheless, RBW is not without limitations. In particular, it depends on models for the conditional means of post-treatment confounders. When these models are incorrectly specified, the pseudo-population generated by residual balancing weights will fail to mimic the original unweighted population, and our estimates will be biased. Even when these models are correctly specified, RBW may also yield biased estimates when we have insufficient balancing conditions. Adding more functions such as cross-products and high-order terms to the set $G(L_t) = \{g_1(L_t), ..., g_{J_t}(L_t)\}$, thereby increasing the number of balancing constraints, may help — though at the risk of making exact balance

infeasible. A possible extension to RBW would be to allow for approximate balance with a penalty for the remaining imbalance in the optimization problem (Fong et al., 2018). Finally, we have relied on the "sandwich" variance estimator to compute standard errors. Though Zhou and Wodtke (2020) demonstrate by simulation studies that this estimator is likely conservative for RBW, the method does not provide a variance estimator tailored to addressing the estimation uncertainty of the RBW weights.

Despite these limitations, RBW will be useful to many social scientists interested in using marginal structural models to study causality in dynamic settings. We will continue to upgrade the package by expanding RBW's ranges of applicability — specifically, to censored data, to contexts involving repeated outcome measures such as survival data (Hernán et al., 2002, 2000), and to cases where, as discussed above, exact balance is infeasible and approximate balance must be pursued.

## Bibliography

M. Blackwell. A Framework for Dynamic Causal Inference in Political Science. 57(2):504–520, 2013. ISSN 0092-5853. doi: 10.1111/j.1540-5907.2012.00626.x. [p1, 13]

S. R. Cole and M. A. Hernán. Constructing Inverse Probability Weights for Marginal Structural Models. *American Journal of Epidemiology*, 168(6):656–664, 2008. ISSN 0002-9262. doi: 10.1093/aje/kwn164. [p1]

C. Fong, C. Hazlett, and K. Imai. Covariate balancing propensity score for a continuous treatment: Application to the efficacy of political advertisements. *The Annals of Applied Statistics*, 12(1), 2018. ISSN 1932-6157. doi: 10.1214/17-AOAS1101. [p1, 4, 9, 16, 17]

C. Fong, M. Ratkovic, and K. Imai. *CBPS: Covariate Balancing Propensity Score*, 2021. URL https://CRAN.R-project.org/package=CBPS. R package version 0.22. [p2]

R. B. Geskus and W. M. van der Wal. *ipw: Estimate Inverse Probability Weights*, 2015. URL https://CRAN.R-project.org/package=ipw. R package version 1.0-11. [p1]

J. Hainmueller. Entropy Balancing for Causal Effects: A Multivariate Reweighting Method to Produce Balanced Samples in Observational Studies. *Political Analysis*, 20(1):25–46, 2012. ISSN 1047-1987. doi: 10.1093/pan/mpr025. [p2, 3, 6]

J. Hainmueller. *ebal: Entropy reweighting to create balanced samples*, 2014. URL https://CRAN.R-project.org/package=ebal. R package version 0.1-6. [p6]

M. Á. Hernán, B. Brumback, and J. M. Robins. Marginal Structural Models to Estimate the Causal Effect of Zidovudine on the Survival of HIV-Positive Men. *Epidemiology (Cambridge, Mass.)*, 11(5): 561–570, 2000. ISSN 1044-3983. doi: 10.1097/00001648-200009000-00012. [p17]

M. A. Hernán, B. A. Brumback, and J. M. Robins. Estimating the causal effect of zidovudine on CD4 count with a marginal structural model for repeated measures. *Statistics in medicine*, 21(12): 1689–1709, 2002. ISSN 0277-6715. doi: 10.1002/sim.1144. [p17]

K. Imai and I. S. Kim. When Should We Use Unit Fixed Effects Regression Models for Causal Inference with Longitudinal Data? *American Journal of Political Science*, 63(2):467–490, 2019. ISSN 0092-5853. doi: 10.1111/ajps.12417. [p13]

K. Imai and M. Ratkovic. Covariate Balancing Propensity Score. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 76(1):243–263, 2014. ISSN 1369-7412. doi: 10.1111/rssb.12027. [p1]

K. Imai and M. Ratkovic. Robust Estimation of Inverse Probability Weights for Marginal Structural Models. *Journal of the American Statistical Association*, 110(511):1013–1023, 2015. ISSN 0162-1459. doi: 10.1080/01621459.2014.956872. [p1, 13]

K. Imai, L. Keele, and T. Yamamoto. Identification, Inference and Sensitivity Analysis for Causal Mediation Effects. *Statistical science*, 25(1):51–71, 2010. ISSN 0883-4237. doi: 10.1214/10-STS321. [p11]

K. Imai, L. Keele, D. Tingley, and T. Yamamoto. Unpacking the Black Box of Causality: Learning about Causal Mechanisms from Experimental and Observational Studies. *The American political science review*, 105(4):765–789, 2011. ISSN 0003-0554. doi: 10.1017/S0003055411000414. [p11]

E. H. Kennedy. *npal: Nonparametric causal inference methods*, 2021. URL https://github.com/ehkennedy/npcausal/blob/master/npcausal.pdf. R package version 0.1.0. [p2]

E. H. Kennedy, Z. Ma, M. D. McHugh, and D. S. Small. Non-parametric methods for doubly robust estimation of continuous treatment effects. *Journal of the Royal Statistical Society. Series B, Statistical methodology*, 79(4):1229–1245, 2017. ISSN 1369-7412. doi: 10.1111/rssb.12212. [p2]

R. R. Lau, L. Sigelman, and I. B. Rovner. The Effects of Negative Political Campaigns: A Meta-Analytic Reassessment. 69(4):1176–1209, 2007. ISSN 0022-3816. doi: 10.1111/j.1468-2508.2007.00618.x. [p1, 13]

A. I. Naimi, E. E. Moodie, N. Auger, and J. S. Kaufman. Constructing Inverse Probability Weights for Continuous Exposures: A Comparison of Methods. *Epidemiology (Cambridge, Mass.)*, 25(2):292–299, 2014. ISSN 1044-3983. doi: 10.1097/EDE.0000000000000053. [p1]

J. M. Robins. A New Approach to Causal Inference in Mortality Studies with a Sustained Exposure Period — Application to Control of the Healthy Worker Survivor Effect. *Mathematical Modelling*, 7 (9):1393–1512, 1986. ISSN 0270-0255. doi: 10.1016/0270-0255(86)90088-6. [p1, 3]

J. M. Robins. Marginal Structural Models versus Structural Nested Models as Tools for Causal Inference. In E. M. Halloran and D. Berry, editors, *Statistical Models in Epidemiology, the Environment, and Clinical Trials*, pages 95–133. Springer, New York, 2000. [p1]

J. M. Robins, M. A. Hernan, and B. Brumback. Marginal Structural Models and Causal Inference in Epidemiology. *Epidemiology*, 11(5):550–560, 2000. ISSN 1044-3983. doi: 10.1097/00001648-200009000-00011. [p1]

M. R. Tomz and J. L. P. Weeks. Public Opinion and the Democratic Peace. 107(4):849–865, 2013. ISSN 0003-0554. doi: 10.1017/S0003055413000488. [p1, 11]

C. Urban and S. Niebler. Dollars on the Sidewalk: Should U.S. Presidential Candidates Advertise in Uncontested States? *American Journal of Political Science*, 58(2):322–336, 2014. ISSN 0092-5853. doi: 10.1111/ajps.12073. [p9, 10]

W. M. van der Wal and R. B. Geskus. Ipw: An R Package for Inverse Probability Weighting. *Journal of Statistical Software*, 43(13):1–23, 2011. ISSN 1548-7660. [p1]

S. Vansteelandt. Estimating Direct Effects in Cohort and Case–Control Studies. *Epidemiology (Cambridge, Mass.)*, 20(6):851–860, 2009. ISSN 1044-3983. doi: 10.1097/EDE.0b013e3181b6f4c9. [p1]

Y. Wang, M. Petersen, D. Bangsberg, and M. van der Laan. Diagnosing Bias in the Inverse Probability of Treatment Weighted Estimator Resulting from Violation of Experimental Treatment Assignment. *U.C. Berkeley Division of Biostatistics Working Paper Series*, Working Paper 211, Sept. 2006. [p1]

X. Zhou and G. T. Wodtke. Residual Balancing: A Method of Constructing Weights for Marginal Structural Models. *Political Analysis*, 28(4):487–506, 2020. ISSN 1047-1987. doi: 10.1017/pan.2020.2. [p1, 2, 3, 6, 10, 11, 13, 14, 15, 16, 17]

*Derick Baum*
*Harvard University*
*Department of Sociology*
derick_baum@g.harvard.edu

*Xiang Zhou*
*Harvard University*
*Department of Sociology*
xiang_zhou@fas.harvard.edu