

# Implementing `blopmatching` in Stata

Juan D. Díaz

Department of Management Control and Information Systems  
University of Chile  
Santiago, Chile  
juadiaz@fen.uchile.cl

Iván Gutiérrez

Department of Statistics  
Pontifical Catholic University of Chile  
Santiago, Chile  
isgutierrez@mat.uc.cl

Jorge Rivera

Department of Economics  
University of Chile  
Santiago, Chile  
jrivera@econ.uchile.cl

**Abstract.** The `blopmatching` estimator for average treatment effects in observational studies is a nonparametric matching estimator proposed by Díaz, Rau, and Rivera (2015, *Review of Economics and Statistics* 97: 803–812). This approach uses the solutions of linear programming problems to build the weighting schemes that are used to impute the missing potential outcomes. In this article, we describe `blopmatch`, a new command that implements these estimators.

**Keywords:** `st0632`, `blopmatch`, average treatment effects, matching, linear programming, synthetic covariate

## 1 Introduction

In this article, we provide an introduction to the “`blopmatching`” estimators for average treatment effects proposed by Díaz, Rau, and Rivera (2015) and describe `blopmatch`, a new command that implements these estimators.

The `blopmatching` approach imputes the missing potential outcome to each unit as a weighted average of observed outcomes of the units with opposite treatment, where the vector of weights is the solution of a nested pair of optimization problems. The first of these optimization problems looks for the weighting schemes that build the “synthetic covariate” defined by Abadie, Diamond, and Hainmueller (2010). Geometrically, the synthetic covariate is the projection of the covariate vector of the unit that needs imputation onto the convex hull of covariate values of the units with opposite treatment.

However, although the synthetic covariate is unique, this first optimization problem often has multiple solutions because there might be several weighting schemes building the synthetic covariate. To overcome this issue of multiplicity of solutions, the “second optimization problem” proposed by Díaz, Rau, and Rivera (2015) implements a refinement criterion for choosing one of the weighting schemes that solves the first optimization problem. More precisely, it chooses the solution that uses the covariate values of units with opposite treatment that are as close as possible to the covariate

value of the unit needing imputation. Thus, blopmatching selects the weighting scheme that maximizes the unit-level covariate balance and, simultaneously, uses the units with opposite treatment with closest covariate values.

Because in the blopmatching approach the counterfactual units and the weighting scheme are determined by solving an optimization problem, it does not need an arbitrary rule to fix these parameters *ex ante*, a crucial aspect of most of the nonparametric approaches currently available in the literature.<sup>1</sup> Finally, as Díaz, Rau, and Rivera (2015) point out, the two optimization problems involved in the blopmatching approach can be collapsed into a single linear programming program, which allows us to implement this estimator by using standard linear programming techniques.

This article is organized as follows. Section 2 presents the blopmatching estimator of average treatment effects where, in addition to presenting the general framework, a simple example illustrates what the blopmatching estimator does. Section 3 shows the procedure to estimate the marginal variances of the blopmatching estimator of average treatment effects. Section 4 details the specific method to solve the aforementioned linear program. Section 5 explains the `blopmatch` command, which implements the blopmatching estimators. Finally, section 6 applies `blopmatch` to the well-known National Supported Work Demonstration, data originally analyzed by LaLonde (1986) and later used by Dehejia and Wahba (1999).

## 2 Blopmatching estimator of average treatment effects

### 2.1 A motivating example

A simple example may be illustrative of what blopmatching does. Suppose that there are three units that are exposed to the control treatment (“control” units), whose observed outcomes are  $Y_1 = Y_1(0)$ ,  $Y_2 = Y_2(0)$ ,  $Y_3 = Y_3(0)$  and whose real valued covariates (continuous) are  $X_1$ ,  $X_2$ , and  $X_3$ , such that  $X_1 < X_2 < X_3$ , so that the convex hull of these covariates is the closed interval  $[X_1, X_3] \subset \mathbb{R}$ . Moreover, suppose that a unit that is exposed to the active treatment (“treated” unit) has observed outcome  $Y_4 = Y_4(1) \in \mathbb{R}$  and covariate  $X_4 \in \mathbb{R}$ . Without loss of generality, we can assume that  $X_4 \neq X_j$ ,  $j = 1, 2, 3$ . Here the goal is to estimate the treatment effect on this fourth unit, that is,  $Y_4(1) - Y_4(0)$ , but the crucial complication is that  $Y_4(0)$ , the outcome of the treated unit under control treatment, is not observed. How does the blopmatching estimator impute the missing potential outcome  $Y_4(0)$ ?

Let  $\text{Proj}(X_4)$  be the projection of  $X_4$  onto the closed interval  $[X_1, X_3]$ , that is, the nearest element of that interval to  $X_4$ . That projection is the synthetic covariate as defined by Abadie, Diamond, and Hainmueller (2010). Notice that when  $X_4 < X_1$ , then  $\text{Proj}(X_4) = X_1$ ; when  $X_3 < X_4$ , then  $\text{Proj}(X_4) = X_3$ ; and when  $X_4 \in [X_1, X_3]$ , then  $\text{Proj}(X_4) = X_4$ .

---

1. For instance, a popular method to estimate treatment effects in observational studies is the nearest-neighbor matching (see Abadie and Imbens [2006]), where the number of neighbors used to impute the missing potential outcome is a tuning parameter defined by the researcher and the weights are equal to the reciprocal of this fixed number.

Regardless of the specific value of the projection, we notice now that the convex combinations of covariate values of control units generating  $\text{Proj}(X_4)$  use vectors of weights  $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)^t \in \mathbb{R}_+^3$  that solve the following system of linear equations:

$$\begin{aligned}\lambda_1 + \lambda_2 + \lambda_3 &= 1 \\ \lambda_1 X_1 + \lambda_2 X_2 + \lambda_3 X_3 &= \text{Proj}(X_4)\end{aligned}\tag{1}$$

Because in general there are multiple vectors  $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)^t \in \mathbb{R}_+^3$  solving this linear system of equations, the solution is actually a “solution set”. Observe also that this solution set is, indeed, the solution of the next optimization problem:

$$\begin{aligned}\min_{(\lambda_1, \lambda_2, \lambda_3)^t \in \mathbb{R}_+^3} & \left| X_4 - \sum_{j=1}^3 \lambda_j X_j \right| \\ \text{s.t.} & \\ & \lambda_1 + \lambda_2 + \lambda_3 = 1\end{aligned}\tag{2}$$

Problem (2) is the first optimization problem used in the blopmatching estimator. Note that, with any weighting scheme solving this problem, we are attaining the best possible covariate balance or, alternatively, we are explaining the covariate of the treated unit through the covariates of the control units as best as possible. However, because there are more than one weighting scheme attaining the best possible covariate balance, a refinement criterion should be implemented to choose one of them. In doing this, we notice that any vector  $\boldsymbol{\lambda}$  in that “solution set” uses covariate  $X_j$ ,  $j = 1, 2, 3$ , whenever  $\lambda_j > 0$ . Thus, the weighted sum of distances (squared) between  $X_4$  and the covariates of the control units is given by

$$\sum_{j=1}^3 \lambda_j |X_4 - X_j|^2$$

The second optimization proposed by the blopmatching aims to minimize this function among the solutions of the first optimization. Hence, using (1) we see that this problem can be posed as the next linear program in terms of weights:

$$\begin{aligned}\min_{(\lambda_1, \lambda_2, \lambda_3)^t \in \mathbb{R}_+^3} & \sum_{j=1}^3 \lambda_j |X_4 - X_j|^2 \\ \text{s.t.} & \\ & \lambda_1 X_1 + \lambda_2 X_2 + \lambda_3 X_3 = \text{Proj}(X_4) \\ & \lambda_1 + \lambda_2 + \lambda_3 = 1\end{aligned}\tag{3}$$

It is clear that when  $X_4 < X_1$ , then the solution of (3) is  $\boldsymbol{\lambda}^b = (1, 0, 0)^t$ , while when  $X_3 < X_4$ , that solution is  $\boldsymbol{\lambda}^b = (0, 0, 1)^t$ . However, when  $\text{Proj}(X_4) = X_4$ , the solution

depends on whether  $X_1 < X_4 < X_2$  or  $X_2 < X_4 < X_3$ ; specifically, the solution of that optimization problem is given by  $\lambda^b = (\lambda_1^b, \lambda_2^b, \lambda_3^b)^t \in \mathbb{R}_+^3$  such that

$$\begin{cases} \lambda_1^b = \frac{X_4 - X_1}{X_2 - X_1}, & \lambda_2^b = \frac{X_2 - X_4}{X_2 - X_1}, & \lambda_3^b = 0 & \text{if } X_1 < X_4 < X_2 \\ \lambda_1^b = 0, & \lambda_2^b = \frac{X_3 - X_4}{X_3 - X_2}, & \lambda_3^b = \frac{X_4 - X_2}{X_3 - X_2} & \text{if } X_2 < X_4 < X_3 \end{cases}$$

Thus, the blopmatching estimator imputes the missing potential outcome of the treated unit as

$$\widehat{Y}_4^b(0) = \lambda_1^b Y_1 + \lambda_2^b Y_2 + \lambda_3^b Y_3 = \begin{cases} Y_1 & \text{if } X_4 < X_1, \\ Y_3 & \text{if } X_4 > X_3 \\ \left(\frac{X_4 - X_1}{X_2 - X_1}\right) Y_1 + \left(\frac{X_2 - X_4}{X_2 - X_1}\right) Y_2 & \text{if } X_1 < X_4 < X_2 \\ \left(\frac{X_3 - X_4}{X_3 - X_2}\right) Y_2 + \left(\frac{X_4 - X_2}{X_3 - X_2}\right) Y_3 & \text{if } X_2 < X_4 < X_3 \end{cases}$$

Lastly, the blopmatching estimator of the unit-level treatment effect that we wanted to estimate is given by  $Y_4 - \widehat{Y}_4^b(0)$ . In the next section, we present the extension of this simple case to a more general setting when covariates are of higher dimension.

## 2.2 General framework

The binary program to be evaluated is defined by the collection

$$\Omega_N = \{(\mathbf{X}_i, Y_i, W_i) \in \mathbb{R}^k \times \mathbb{R} \times \{0, 1\} : i \in \{1, \dots, N\}\}$$

where  $N$  is the number of units,  $\mathbf{X}_i \in \mathbb{R}^k$  denotes the observed  $k$ -dimensional vector of covariates (or pretreatment variables) of unit  $i \in \{1, \dots, N\}$ , whose observed outcome is  $Y_i \in \mathbb{R}$ , and  $W_i \in \{0, 1\}$  indicates the treatment received by this unit (active treatment when  $W_i = 1$ , control treatment if  $W_i = 0$ ). The number of treated units is  $N_1 = \sum_{i=1}^N W_i$ , and the number of control units is  $N_0 = N - N_1$ . For the sake of simplifying the exposition of the method, the next nonrestrictive conditions are assumed throughout the article.

- i)  $N_0 > 1$  and  $N_1 > 1$ ,
- ii)  $W_i = 0$  for  $i \in \{1, \dots, N_0\}$ , and  $W_i = 1$  for  $i \in \{N_0 + 1, \dots, N\}$ .

Condition ii states that the first  $N_0$  units of the sample are control units, while the remaining  $N_1 = N - N_0$  are the treated ones. Under condition ii, the covariate vectors of the control units are  $\mathbf{X}_1, \dots, \mathbf{X}_{N_0}$ , whereas the covariate vectors of the treated units are  $\mathbf{X}_{N_0+j}$ ,  $j = 1, \dots, N_1$ .

We are interested in estimating the average treatment effect (ATE) of the program, that is,  $\tau = \mathbb{E}\{Y_i(1) - Y_i(0)\}$ , and the average treatment effect on the treated (ATT), that is,  $\tau_{\text{tre}} = \mathbb{E}\{Y_i(1) - Y_i(0) | W_i = 1\}$ , where  $Y_i(0)$  denotes the outcome of unit  $i$  under

control treatment and  $Y_i(1)$  denotes the outcome of unit  $i$  under active treatment. In this context,  $Y_i(0)$  and  $Y_i(1)$  are called “potential outcomes” (Rubin 1974) because, for each unit, we never observe both of them in the data at hand. That is,  $Y_i(0)$  is not observed if  $W_i = 1$  and  $Y_i(1)$  is not observed if  $W_i = 0$ . Thus, the fundamental problem of causal inference is that the missing potential outcomes must be estimated (or imputed) to estimate either the ATE or the ATT.

Throughout, the Simplex of dimension  $n \in \mathbb{N}$  is denoted as  $\Delta_n = \{(\lambda_1, \dots, \lambda_n)^t \in \mathbb{R}_+^n : \lambda_1 + \dots + \lambda_n = 1\}$ . Having defined the Simplex, in this article, we are interested in the estimator of  $Y_i(1 - W_i)$ , the missing potential outcome of unit  $i$ , of the form

$$\widehat{Y}_i(1 - W_i) = \begin{cases} \sum_{j=1}^{N_1} \lambda_{i,j} Y_{N_0+j} & \text{if } i \in \{1, \dots, N_0\} \\ \sum_{j=1}^{N_0} \lambda_{i,j} Y_j & \text{if } i \in \{N_0 + 1, \dots, N\} \end{cases}$$

with  $\lambda_i = (\lambda_{i,1}, \dots, \lambda_{i,N_1})^t \in \Delta_{N_1}$  when  $W_i = 0$  and  $\lambda_i = (\lambda_{i,1}, \dots, \lambda_{i,N_0})^t \in \Delta_{N_0}$  when  $W_i = 1$ .

For a treated unit  $i \in \{N_0 + 1, \dots, N\}$  whose  $Y_i(0)$  needs to be estimated, the blopmatching method uses the vector of weights that solves the next linear programming program, a straightforward extension of (3) to a higher dimension of the vector of covariates,

$$\begin{aligned} & \min_{(\lambda_1, \dots, \lambda_{N_0})^t \in \mathbb{R}_+^{N_0}} \sum_{j=1}^{N_0} \lambda_j \|\mathbf{X}_i - \mathbf{X}_j\|^2 \\ & \text{s.t.} \\ & \sum_{j=1}^{N_0} \lambda_j \mathbf{X}_j = \text{Proj}(\mathbf{X}_i) \\ & \sum_{j=1}^{N_0} \lambda_j = 1 \end{aligned} \tag{4}$$

where  $\|\cdot\|$  denotes a given norm in the underlying vector space and  $\text{Proj}(\mathbf{X}_i) \in \mathbb{R}^k$  is the projection of  $\mathbf{X}_i$  onto the convex hull of covariate values of control units. It is worth recalling that the convex hull of vectors  $\mathbf{X}_1, \dots, \mathbf{X}_{N_0}$  is the subset conformed by all their convex combinations. This is denoted as  $\text{conv}\{\mathbf{X}_1, \dots, \mathbf{X}_{N_0}\} = \{\lambda_1 \mathbf{X}_1 + \dots + \lambda_{N_0} \mathbf{X}_{N_0} : (\lambda_1, \dots, \lambda_{N_0})^t \in \Delta_{N_0}\} \subset \mathbb{R}^k$ . Note that  $\text{Proj}(\mathbf{X}_i)$  is the unique vector that complies with

$$\|\mathbf{X}_i - \text{Proj}(\mathbf{X}_i)\| = \min_{\mathbf{X} \in \text{conv}\{\mathbf{X}_1, \dots, \mathbf{X}_{N_0}\}} \|\mathbf{X}_i - \mathbf{X}\| \tag{5}$$

Let  $\lambda_i^b = (\lambda_{i,1}^b, \dots, \lambda_{i,N_0}^b)^t \in \Delta_{N_0}$  denote the solution of (4), which under continuous covariates is unique with probability one. Then, the blopmatching estimator of the missing potential outcome of treated unit  $i$ ,  $Y_i(0)$ , is

$$\widehat{Y}_i^b(0) = \sum_{j=1}^{N_0} \lambda_{i,j}^b Y_j$$

A completely analogous procedure can be presented when the unit needing imputation belongs to the control group, that is, when  $i \in \{1, \dots, N_0\}$  and  $Y_i(1)$  needs to be imputed. Specifically, for that case,  $\Delta_{N_0}$ ,  $\{1, \dots, N_0\}$ , and  $\text{Proj}(\mathbf{X}_i)$  in (4) must be replaced, respectively, by  $\Delta_{N_1}$ ,  $\{N_0 + 1, \dots, N\}$ , and the projection of  $\mathbf{X}_i$  onto the convex hull of  $\{\mathbf{X}_{N_0+1}, \dots, \mathbf{X}_N\}$ . Let  $\lambda_i^b = (\lambda_{i,1}^b, \dots, \lambda_{i,N_1}^b)^t \in \Delta_{N_1}$  denote the solution of the optimization problem configured for a control unit  $i$ . Then, the blopmatching estimator of its missing potential outcome,  $Y_i(1)$ , is

$$\widehat{Y}_i^b(1) = \sum_{j=1}^{N_1} \lambda_{i,j}^b Y_{N_0+j}$$

Having imputed all the missing potential outcomes, we define the blopmatching estimators of the ATE and the ATT as follows:

**Definition 1.** The blopmatching estimators of  $\tau$  (ATE) and  $\tau_{\text{tre}}$  (ATT) are, respectively,

$$\begin{aligned} \widehat{\tau}^b &= \frac{1}{N} \sum_{i=1}^N \left[ W_i \left\{ Y_i - \widehat{Y}_i^b(0) \right\} + (1 - W_i) \left\{ \widehat{Y}_i^b(1) - Y_i \right\} \right] \\ \widehat{\tau}_{\text{tre}}^b &= \frac{1}{N_1} \sum_{i=1}^{N_1} \left\{ Y_i - \widehat{Y}_i^b(0) \right\} \end{aligned}$$

We highlight that, to impute the missing potential outcome of any unit, the blopmatching approach does not exclude a priori any unit with opposite treatment. We also highlight that the blopmatching approach does not need to fix an exogenous number of neighbors, a crucial tuning parameter in most nonparametric approaches (see Imbens and Wooldridge [2009]).

We note now that for a treated unit  $i \in \{N_0 + 1, \dots, N\}$ , the observed outcome of control unit  $j \in \{1, \dots, N_0\}$  participates in the realization of  $\widehat{Y}_i^b(0)$  whenever  $\lambda_{i,j}^b$  is strictly positive. Because the number of equality constrains of (4) is  $k + 1$ , it follows that a “basic optimal solution” of that problem will have, at most,  $k + 1$  strictly positive components. Consequently, the number of control units whose outcomes are used when the blopmatching estimator imputes  $Y_i(0)$  should be at most  $k + 1$ . We observe, however, that these units should not be necessarily the first  $k + 1$  nearest neighbors to unit  $i$  (in terms of covariate distances); indeed, they might not even be among the first  $M$ -nearest neighbors for some fixed integer  $M$ . For instance, let  $k = 1$ , and suppose that a treated

unit with covariate  $X_i = 1 - \epsilon$ , with  $0 < \epsilon < 1$ , has to be matched against control units with covariate values  $X_1 = 0, X_2 = 1$  and  $X_j = 1 + 1/j, j = 3, \dots, N_0$ . In this case,  $\boldsymbol{\lambda}_i^b = (\epsilon, 1 - \epsilon, 0, \dots, 0) \in \boldsymbol{\Delta}_{N_0}$  is the solution of (4), while for  $\epsilon$  small enough, the nearest neighbors to unit  $i$  are in order of closeness:  $X_2, X_{N_0}, X_{N_0-1}, \dots, X_3, X_1$ . Hence, the units used to impute the missing potential outcome of  $X_i$  are its nearest and farthest neighbors.

We end this section saying that, under the assumption of strong ignorability and some additional mild conditions, Díaz, Rau, and Rivera (2015) prove that the blopmatching estimators of the ATE and ATT are consistent. We recall that the assumption of strong ignorability holds if two conditions are satisfied: unconfoundedness, which means that the treatment assignment and the potential outcomes are conditional independent given covariates (this condition is also known as selection on observed covariates); and positivity, which means that the conditional probability of being treated given covariates is strictly above zero and strictly below one (this condition is also known as overlap in the propensity score). See chapter 12 in Imbens and Rubin (2015) for a detailed discussion on observational studies under strong ignorability and chapter 21 for a procedure to assess the plausibility of this assumption in practice.

### 3 Estimating the marginal variance

Díaz, Rau, and Rivera (2015) propose consistent estimators of the variances of  $\hat{\tau}^b$  and  $\hat{\tau}_{\text{tre}}^b$ , whose expressions use an estimator of the conditional variance  $\sigma^2(\mathbf{x}) = \mathbb{V}(Y_i | \mathbf{X}_i = \mathbf{x}, W_i = w)$ . The estimator of  $\sigma^2(\mathbf{x})$  relies on the solution of a similar optimization problem to (4). For any unit  $i$ , instead of using covariates of units belonging to the group with opposite treatment, this problem uses covariates of units in the group with the same treatment as the unit  $i$  under analysis, but of course leaving out the covariate vector of unit  $i$ . More precisely, for a treated unit  $i \in \{N_0 + 1, \dots, N\}$ , we denote  $\boldsymbol{\varphi}_{-i}^b = (\varphi_1^b, \dots, \varphi_{i-1}^b, \varphi_{i+1}^b, \dots, \varphi_{N_1}^b)^t \in \boldsymbol{\Delta}_{N_1-1}$  the solution of the following linear optimization problem,

$$\begin{aligned} & \min_{(\varphi_1, \dots, \varphi_{i-1}, \varphi_{i+1}, \dots, \varphi_{N_1})^t \in \mathbb{R}_+^{N_1-1}} \sum_{\substack{j=1 \\ j \neq i}}^{N_1} \varphi_j \|\mathbf{X}_i - \mathbf{X}_{N_0+j}\|^2 \\ & \text{s.t.} \\ & \sum_{\substack{j=1 \\ j \neq i}}^{N_1} \varphi_j \mathbf{X}_{N_0+j} = \text{Proj}^*(\mathbf{X}_i) \\ & \sum_{j=1, j \neq i}^{N_1} \varphi_j = 1 \end{aligned} \tag{6}$$

where  $\text{Proj}^*(\mathbf{X}_i)$  is the projection of  $\mathbf{X}_i$  onto  $\text{conv}\{\mathbf{X}_{N_0+1}, \dots, \mathbf{X}_{i-1}, \mathbf{X}_{i+1}, \dots, \mathbf{X}_N\}$ . It can be shown (see Díaz, Rau, and Rivera [2015]) that the next expression is an asymptotically unbiased estimator of the conditional variance,

$$\hat{\sigma}_i^2 = \frac{\left\{ Y_i - \left( \sum_{j=1, j \neq i}^{N_1} \varphi_j^b Y_{N_0+j} \right) \right\}^2}{1 + \|\boldsymbol{\varphi}_{-i}^b\|_2^2}$$

where  $\|\cdot\|_2$  is the Euclidean norm in  $\mathbb{R}^{N_1-1}$ .

For  $i \in \{1, \dots, N_0\}$ , a completely analogous procedure can be implemented to obtain an estimator of the conditional variance  $\sigma^2(\mathbf{X}_i)$ . Having estimated all the conditional variances, we find the following expressions to be consistent estimators (see Díaz, Rau, and Rivera [2015] for a detailed proof and the required conditions) of the marginal variances of the blopmatching estimators of the ATE and ATT,

$$\begin{aligned} \hat{V}[\hat{\tau}^b] &= \frac{1}{N^2} \sum_{i=1}^N \left[ \left( Y_i - \hat{Y}_i^b \right)^2 + \left\{ \left( 1 + c_i^{[1]} \right)^2 - \left( 1 + c_i^{[2]} \right) \right\} \hat{\sigma}_i^2 \right] - \frac{(\hat{\tau}^b)^2}{N} \\ \hat{V}[\hat{\tau}_{\text{tre}}^b] &= \frac{1}{N_1^2} \sum_{i=N_1+1}^N \left( Y_i - \hat{Y}_i^b \right)^2 + \frac{1}{N_1^2} \sum_{i=1}^{N_0} \left\{ \left( c_i^{[1]} \right)^2 - c_i^{[2]} \right\} \hat{\sigma}_i^2 - \frac{(\hat{\tau}_{\text{tre}}^b)^2}{N_1} \end{aligned}$$

where

$$c_i^{[s]} = \sum_{j: W_j \neq W_i} (\lambda_{j,i}^b)^s \quad s = 1, 2$$

that is, the sum of weights, to the power of  $s$ , corresponding to unit  $i$  when  $i$  is used to impute the missing potential outcomes of units in the group with opposite treatment.

## 4 Implementation

This section presents our strategy to solve (4). Because of the similar nature of both (4) and (6), this procedure can be accommodated to solve the linear programs given by (6) and estimate the marginal variances of the treatment-effect estimators.

To solve (4), we first need to obtain the projection of  $\mathbf{X}_i$  onto the convex hull of covariate values of units with opposite treatment condition (that is, with treatment equal to  $1 - W_i$ ). For the sake of presentation, we initially suppose that  $i \in \{N_0 + 1, \dots, N\}$  and  $i$  is a treated unit; for further convenience that will be clear later on in this section, we use Norm 1 (denoted as  $\|\cdot\|_1$ ) to solve the projection problem given by (5). Here it is worth recalling that, for  $\mathbf{X} = (x_1, \dots, x_n)^t \in \mathbb{R}^n$ ,  $\|\mathbf{X}\|_1 = |x_1| + \dots + |x_n|$ . In view of these choices, we can see that (5) becomes a linear programming problem. Notice now that  $\text{Proj}(\mathbf{X}_i)$  is performed by the weighting schemes that solve the next optimization problem:

$$\min_{\boldsymbol{\lambda}=(\lambda_1,\dots,\lambda_{N_0})^t \in \Delta_{N_0}} \left\| \mathbf{X}_i - \left( \sum_{j=1}^{N_0} \lambda_j \mathbf{X}_j \right) \right\|_1 \quad (7)$$

Let  $\mathbf{c}_i = \mathbf{X}_i - \text{Proj}(\mathbf{X}_i) \in \mathbb{R}^k$ , and let  $\boldsymbol{\lambda}_i^* \in \Delta_{N_0}$  denote a solution to (7). Then, it is clear that  $(\mathbf{c}_i, \boldsymbol{\lambda}_i^*)^t \in \mathbb{R}^k \times \Delta_{N_0}$  is a solution to the next optimization problem:

$$\begin{aligned} & \min_{(\mathbf{c}, \boldsymbol{\lambda})^t \in \mathbb{R}^k \times \Delta_{N_0}} \|\mathbf{c}\|_1 \\ & \text{s.t.} \\ & \mathbf{c} + \sum_{j=1}^{N_0} \lambda_j \mathbf{X}_j = \mathbf{X}_i \end{aligned}$$

Denoting  $\boldsymbol{\iota}_k = (1, \dots, 1)^t \in \mathbb{R}^k$  and  $|\mathbf{c}| = (|\mathbf{c}_1|, \dots, |\mathbf{c}_k|)^t \in \mathbb{R}_+^k$ , we see that it follows directly that  $\|\mathbf{c}\|_1 = \boldsymbol{\iota}_k^t |\mathbf{c}|$ . Moreover, it is also clear that there are a couple of vectors  $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}_+^k$  such that

$$\mathbf{c} = |\mathbf{c}| - 2\boldsymbol{\beta} \quad \text{and} \quad \mathbf{c} = -|\mathbf{c}| + 2\boldsymbol{\alpha}$$

implying that  $\mathbf{c} = \boldsymbol{\alpha} - \boldsymbol{\beta}$  and  $|\mathbf{c}| = \boldsymbol{\alpha} + \boldsymbol{\beta}$ . Consequently,  $\|\mathbf{c}\|_1 = \boldsymbol{\iota}_k^t (\boldsymbol{\alpha} + \boldsymbol{\beta})$ , and then

$$\text{Proj}(\mathbf{X}_i) = \mathbf{X}_i - (\boldsymbol{\alpha}_i^* - \boldsymbol{\beta}_i^*)$$

with  $\boldsymbol{\alpha}_i^*, \boldsymbol{\beta}_i^* \in \mathbb{R}^k$  as the corresponding components of a solution to the next linear programming problem,

$$\begin{aligned} & \min_{(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda})^t \in \mathbb{R}_+^k \times \mathbb{R}_+^k \times \mathbb{R}_+^{N_0}} \boldsymbol{\iota}_k^t \boldsymbol{\alpha} + \boldsymbol{\iota}_k^t \boldsymbol{\beta} \\ & \text{s.t.} \\ & \begin{bmatrix} \mathbf{I}_k & -\mathbf{I}_k & \mathbf{X}_1 & \dots & \mathbf{X}_{N_0} \\ \mathbf{0}_k^t & \mathbf{0}_k^t & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_i \\ 1 \end{bmatrix} \end{aligned} \quad (8)$$

where  $\mathbf{I}_k$  stands for the identity matrix of  $\mathbb{R}^{k \times k}$  and  $\mathbf{0}_k \in \mathbb{R}^k$  is the vector with all entries zero.

In the following, for  $j = 1, \dots, n$ ,  $\mathbf{e}_j^n$  denotes the  $j$ -vector of the canonical basis of  $\mathbb{R}^n$ . Because (8) has  $k+1$  equality constraints, to implement the ‘‘revised Simplex method’’ (see Matoušek and Gärtner [2007]) to obtain a solution, we should ‘‘mark’’  $k+1$  linear independent (l.i) columns of the  $(k+1) \times (2k+N_0)$  matrix on the left-hand side of restrictions of (8) (this matrix denoted as  $\boldsymbol{\Gamma}$ ). In doing this, by setting  $K = \{1, \dots, k\}$  and denoting  $\mathbf{X}_i = (X_{1,i}, \dots, X_{k,i})^t$ , we set

$$\mathcal{B}_i = \{s \in K : X_{s,i} \geq X_{s,N_0}\} \cup \{k+s : s \in K, X_{s,i} < X_{s,N_0}\} \cup \{2k+N_0\}$$

It is clear that the  $(2k+N_0)$ -column of matrix  $\boldsymbol{\Gamma}$ , namely, the vector  $(\mathbf{X}_{N_0}, 1)^t \in \mathbb{R}^{k+1}$ , is l.i from the first  $2k$  columns of that matrix. It is also clear that, by construction, the

columns of  $\mathbf{\Gamma}$  marked by  $\mathcal{B}_i \setminus \{2k + N_0\}$  are pairwise orthogonal. In consequence, columns of  $\mathbf{\Gamma}$  marked by  $\mathcal{B}_i$  conform a basis of  $\mathbb{R}^{k+1}$ . To show these columns give a “basic feasible solution” for (8), we show that a solution induced by them is nonnegative. In this regard, it is easy to check that such a solution is given by  $\bar{\boldsymbol{\alpha}} = (\bar{\alpha}_j) \in \mathbb{R}^k$ ,  $\bar{\boldsymbol{\beta}} = (\bar{\beta}_j) \in \mathbb{R}^k$ , and  $\bar{\boldsymbol{\lambda}} \in \mathbb{R}^{N_0}$  such that

$$\begin{aligned} \bar{\alpha}_j &= + \max\{X_{j,i} - X_{j,N_0}, 0\}, j \in \{1, \dots, k\} \\ \bar{\beta}_j &= - \min\{X_{j,i} - X_{j,N_0}, 0\}, j \in \{1, \dots, k\} \\ \bar{\boldsymbol{\lambda}} &= \mathbf{e}_{N_0}^{N_0} \end{aligned}$$

Hence, by using the basic feasible solution above, we can solve (8) by the revised Simplex method, which gives the vector  $\text{Proj}(\mathbf{X}_i)$ , as desired. All the commands needed for that purpose, programmed using Mata codes, are provided in addition to this article.

Having found  $\text{Proj}(\mathbf{X}_i)$ , we can solve (4). Specifically, our commands implement the solution using either the Euclidean distance or the Mahalanobis distance (the choice is left to the researchers). In what follows, the norm is  $\|\cdot\|$ , and, without loss of generality, we present the implementation for a treated unit  $i \in \{N_0 + 1, \dots, N\}$ .

We observe now that  $\boldsymbol{\lambda}_i^b = (\lambda_{i,1}^b, \dots, \lambda_{i,N_0}^b)^t \in \boldsymbol{\Delta}_{N_0}$ , which solves (4), is also a solution of the next linear optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\lambda} \in \mathbb{R}_+^{N_0}} \quad & \sum_{j=1}^{N_0} \lambda_j \|\mathbf{X}_j - \mathbf{X}_i\|^2 \\ \text{s.t.} \quad & \begin{bmatrix} \mathbf{X}_1 & \dots & \mathbf{X}_{N_0} \\ 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_{N_0} \end{bmatrix} = \begin{bmatrix} \text{Proj}(\mathbf{X}_i) \\ 1 \end{bmatrix} \end{aligned} \tag{9}$$

We observe also that any solution  $\boldsymbol{\lambda}_i^* = (\lambda_{i,1}^*, \dots, \lambda_{i,N_0}^*)^t$  of (7) is a feasible point for (9), although not necessarily a basic solution. Nonetheless, it is not difficult to realize that the subset

$$\mathcal{C}_i = \{j \in \{1, \dots, N_0\} : \lambda_{i,j}^b > 0\}$$

identifies  $\#\mathcal{C} \leq k + 1$  linear independent columns of the coefficient matrix on the left-hand side of restrictions of (9). If we assume that  $N_0 > k$ ,  $N_1 > k$ , and covariates are continuous, under general conditions (see Cover and Efron [1967]), then any  $k + 1$  columns of that matrix are l.i with probability one (these columns are in “general position”). Consequently, without loss of generality, we can assume that  $\#\mathcal{C} = k + 1$ , implying that the vector  $\boldsymbol{\lambda}_i^*$  is a basic feasible solution of (9). Using this vector, we can again implement the revised Simplex method to solve (9).<sup>2</sup> For  $N_0 \leq k$  (or  $N_1 \leq k$ ), (9) can be converted into a linear programming problem with standard form so that the revised Simplex method can be directly used with  $\boldsymbol{\lambda}_i^*$  as a basic feasible solution. To implement the `blomatch` command in Stata, we programmed all the commands involved in solving this optimization problem in Mata codes.

2. For  $\#\mathcal{C} < k + 1$ , we can follow standard approaches to extend subset  $\mathcal{C}_i$  and complete a basis for that problem; see theorem 2.33 in Axler (2014).

## 5 The *blopmatch* command

### 5.1 Syntax

`blopmatch` (*ovar* *omvarlist*) (*tvar*) [*if*] [*in*] [*weight*] [, *stat options*]

*ovar* is a binary, count, continuous, fractional, or nonnegative outcome of interest.

*omvarlist* specifies the covariates in the outcome model. *omvarlist* may contain factor variables; see [U] 11.4.3 **Factor variables**.

*tvar* must contain integer values representing the treatment levels. Only two treatment levels are allowed.

*by* and *statsby* are allowed; see [U] 11.1.10 **Prefix commands**.

*fweights* are allowed; see [U] 11.1.6 **weight**.

<i>stat</i>	Description
<code>ate</code>	estimate average treatment effect in population; the default
<code>atet</code>	estimate average treatment effect on the treated

  

<i>options</i>	Description
<code>ematch(<i>varlist</i>)</code>	match exactly on specified variables
<code>llevel(#)</code>	set confidence level; default is <code>llevel(95)</code>
<code>dmvariables</code>	display names of matching variables
<code>control(# <i>label</i>)</code>	specify the level of <i>tvar</i> that is the control
<code>tlevel(# <i>label</i>)</code>	specify the level of <i>tvar</i> that is the treatment
<code>metric(<i>metric</i>)</code>	select distance metric for covariates
<code>btolerance</code>	solver tolerance in simplex algorithm (boundedness test)
<code>otolerance</code>	solver tolerance in simplex algorithm (optimality test)
<code>iterate</code>	maximum number of iterations

  

<i>metric</i>	Description
<code>mahalanobis</code>	inverse sample covariate covariance; the default
<code>ivariance</code>	inverse diagonal sample covariate covariance
<code>euclidean</code>	identity

## 5.2 Stored results

`blpmatch` stores the following in `e()`:

### Scalars

<code>e(N)</code>	number of observations
<code>e(n<sub>j</sub>)</code>	number of observations for treatment level $j$
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control

### Macros

<code>e(cmd)</code>	<code>blpmatch</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(tvar)</code>	name of treatment variable
<code>e(emvarlist)</code>	exact match variables
<code>e(mvarlist)</code>	match variables
<code>e(metric)</code>	<code>mahalanobis</code> , <code>ivariance</code> , or <code>euclidean</code>
<code>e(stat)</code>	statistic estimated, <code>ate</code> or <code>atet</code>
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(tlevels)</code>	levels of treatment variable
<code>e(properties)</code>	<code>b V</code>

### Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

### Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

## 5.3 Example

We illustrate the use of `blpmatch` to estimate the average treatment effect of the National Supported Work Demonstration, a labor training program, on postintervention earnings, using data from LaLonde (1986):

```
. use nsw
(Sample from the National Supported Work Demonstration)
```

Table 1 describes all variables in this dataset.

Table 1. Description of all variables in `nsw.dta`

Variable	Description
<code>treat</code>	Treatment indicator (1 if treated, 0 otherwise)
<code>age</code>	Age
<code>education</code>	Education
<code>black</code>	1 if black, 0 otherwise
<code>hispanic</code>	1 if Hispanic, 0 otherwise
<code>married</code>	1 if married, 0 otherwise
<code>nodegree</code>	1 if no degree, 0 otherwise
<code>re75</code>	Earnings in 1975 (in 1978 dollars)
<code>re78</code>	Earnings in 1978 (in 1978 dollars)

Now we use `blopmatch` to estimate the average treatment effect of `treat` on `re78`. Subject distances are measured using the Mahalanobis distances defined by covariates `age`, `education`, `black`, `hispanic`, `married`, `nodegree`, and (a standardized version of) `re75`.

```
. egen stdre75 = std(re75)
. local Y = "re78"
. local W = "treat"
. local X = "age education black hispanic married nodegree stdre75"
. blopmatch (`Y' `X') (`W'), ate
```

Treatment-effects estimation

Estimator	: blop matching	Number of obs	=	722
Outcome model	: matching	Control group size	=	425
Distance Metric:	Mahalanobis	Treatment group size	=	297

re78	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
ate	975.6248	503.8534	1.94	0.053	-11.90981 1963.159

Thus, the ATE of the National Supported Work Demonstration on postintervention earnings is 975.6 USD (of 1978). In contrast, the ATE estimated with `teffects nmatch` depends on the number of neighbors. For example,

```
. teffects nmatch (`Y' `X') (`W'), ate nneighbor(01) // ATE = 744.9789
(output omitted)
. teffects nmatch (`Y' `X') (`W'), ate nneighbor(16) // ATE = 968.9167
(output omitted)
```

## 6 Acknowledgments

The authors gratefully acknowledge financial support from ANID PIA/APOYO AFB180003 and from FONDECYT-Chile, grant number 1130468.

## 7 Programs and supplemental materials

To install a snapshot of the corresponding software files as they existed at the time of publication of this article, type

```
. net sj 21-1
. net install st0632      (to install program files, if available)
. net get st0632         (to install ancillary files, if available)
```

You can also install this package from GitHub by typing the following commands:

```
. net install blopmatch,
> from(https://raw.githubusercontent.com/igutierrezm/blopmatch/master/)
```

## 8 References

- Abadie, A., A. Diamond, and J. Hainmueller. 2010. Synthetic control methods for comparative case studies: Estimating the effect of California’s tobacco control program. *Journal of the American Statistical Association* 105: 493–505. <https://doi.org/10.1198/jasa.2009.ap08746>.
- Abadie, A., and G. W. Imbens. 2006. Large sample properties of matching estimators for average treatment effects. *Econometrica* 74: 235–267. <https://doi.org/10.1111/j.1468-0262.2006.00655.x>.
- Axler, S. 2014. *Linear Algebra Done Right*. 3rd ed. Cham, Switzerland: Springer.
- Cover, T. M., and B. Efron. 1967. Geometrical probability and random points on a hypersphere. *Annals of Mathematical Statistics* 38: 213–220. <http://doi.org/10.1214/aoms/1177699073>.
- Dehejia, R. H., and S. Wahba. 1999. Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs. *Journal of the American Statistical Association* 94: 1053–1062. <https://doi.org/10.1080/01621459.1999.10473858>.
- Díaz, J., T. Rau, and J. Rivera. 2015. A matching estimator based on a bilevel optimization problem. *Review of Economics and Statistics* 97: 803–812. [https://doi.org/10.1162/REST\\_a.00504](https://doi.org/10.1162/REST_a.00504).
- Imbens, G. W., and D. B. Rubin. 2015. *Causal Inference in Statistics, Social, and Biomedical Sciences*. Cambridge: Cambridge University Press.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. <https://doi.org/10.1257/jel.47.1.5>.

- LaLonde, R. J. 1986. Evaluating the econometric evaluations of training programs. *American Economic Review* 76: 604–620.
- Matoušek, J., and B. Gärtner. 2007. *Understanding and Using Linear Programming*. Verlag: Springer.
- Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701. <https://doi.org/10.1037/h0037350>.

**About the authors**

Juan D. Díaz is an assistant professor in the Department of Management Control and Information Systems at the Faculty of Economics and Business of the University of Chile.

Iván Gutiérrez is a PhD student in the Department of Statistics at the Pontifical Catholic University of Chile.

Jorge Rivera is an associate professor in the Department of Economics at the Faculty of Economics and Business of the University of Chile and a researcher at Instituto Sistemas Complejos de Ingeniería (ISCI).