blopmatch: blop-matching en Stata

Tesis para optar al grado de
Magíster en Economía

Alumno: Iván Gutiérrez
Profesor Guía: Jorge Rivera
Comisión: Esteban Puentes, Juan Díaz

11 de abril de 2023

blopmatch: blop-matching en Stata

Resumen

En esta tesis se define e introduce el `blopmatch`, un nuevo comando de Stata (13+) para estimar efectos promedio de tratamiento usando *blop-matching* definido por Díaz et al. [2015]. Primero se describe el método de estimación, y luego se presenta la rutina que lo implementa. Además se describe y desarrolla la rutina que permite obtener un estimador consistente de la varianza de los estimadores de efecto promedio de tratamiento.

Notación

Conjuntos

Dados $n \in \mathbb{N}$ y $S \subseteq \mathbb{R}^n$, se definen:

Δ_n	El *n-símplex unitario.*
$\operatorname{co} S$	La envoltura convexa de S.

Matrices

Dados $n, m \in \mathbb{N}$, se definen:

$\boldsymbol{I}_n$	La identidad de orden n.
$\boldsymbol{e}_{nm}$	La n-ésima columna de $\boldsymbol{I}_m$.
$\boldsymbol{\imath}_n$	El vector de unos de orden n.
$\mathbf{0}_n$	El vector de ceros de orden n.

Por defecto, todos los vectores se presumen vectores columna.

Índice general

1

blopmatch: blop-matching en Stata

1 *Introducción*

El *blop-matching* propuesto por Díaz et al. [2015] es un método no paramétrico para estimar, en estudios observacionales, el efecto tratamiento de programas binarios sobre cierta variable de interés (ver Imbens and Wooldridge [2009]). Esta tesis de grado describe el comando `blopmatch` que implementa dicho método en Stata.

A diferencia de otras técnicas de estimación, el *blop-matching* considera el balance de cada unidad bajo análisis para la definición de los ponderadores que serán empleados en la construcción del estimador de efecto del tratamiento a nivel individual.

Una manera de entender los ponderadores que propone el *blop-matching* es que ellos son una extensión del mecanismo de asignación de parejas del *matching* con un vecino más cercano (en términos de distancia entre covariables). Para cada unidad bajo análisis, el vector de ponderadores que se obtiene con el *blop-matching* resuelve un problema de optimización cuya función objetivo es la misma que define a su vecino más cercano, pero donde ahora se añade una restricción que asegure el mejor *balance* posible entre el vector de covariables de esa unidad y aquellas de sus posibles parejas dentro de las unidades que tienen tratamiento opuesto. Como consecuencia del método, para la estimación del efecto tratamiento individual ocurre que tanto el número de unidades con tratamiento opuesto cuyos resultados son empleados en la estimación, como los ponderadores que se utilizan en hacer dicha estimación, se determinan de manera endógena.

El problema de optimización en comento se puede plantear como un problema de programación lineal, donde además de requerir que los ponderadores sean positivos y sumen uno, deben satisfacer un sistema de ecuaciones lineales con lado derecho igual a la proyección del vector de covariables de la unidad bajo análisis sobre la envoltura convexa de las covariables de las unidades con tratamiento opuesto (cuestión que garantiza el mejor balance de covariables ya indicado).

La principal dificultad para implementar `blopmatch` en Stata provino del hecho que dicho software no posee rutinas propias para resolver problemas de optimización (lineal), por lo que fue necesario programar, y probar, todo lo concerniente para determinar las proyecciones mencionadas, y resolver el problema lineal que produce los ponderadores. Eso se hizo de manera eficiente, y según un enfoque propio que tomó en cuenta diversos aspectos de la programación lineal, logrando así rutinas estables y eficaces. Adicionalmente, este trabajo implementa el estimador de la varianza de los estimadores de efecto tratamiento.

Esta tesis se organiza como sigue. En las Secciones 2 y 3 se introducen el marco teórico y el *blop-matching*, respectivamente, mientras que la Sección 4 explica cómo resolver los problemas de optimización que el *blop-matching* plantea. En la Sección 5 se introduce el comando `blopmatch` y, finalmente, en la Sección 6 se presenta un ejemplo de aplicación usando la base de datos de LaLonde [1986].

2 *Marco Teórico*

2.1 *Efectos promedio de tratamiento*

Considere el problema de estimar el efecto causal de un tratamiento binario sobre una variable de resultado. Con el fin de evaluar dicho efecto, supongamos que se cuenta con una muestra de N unidades, y que se observa la siguiente información:

$i \in I := \{1, \ldots, N\}$	su *id* dentro de la muestra,
$X_i = (X_{1i}, \ldots, X_{Ki})' \in \mathbb{R}^K$	vector de covariables de unidad i,
$W_i \in \{0, 1\}$	nivel de tratamiento de unidad i,
$Y_i \in \mathbb{R}$	variable de resultado de unidad i.

En lo anterior, se asume que $W_i = 1$ si la unidad $i \in \{1, \ldots, N\}$ recibe el tratamiento, y $W_i = 0$ en caso contrario (unidad de control). Sin pérdida de generalidad, se asume que las primeras N_0 unidades pertenecen al grupo de control, mientras que las $N_1 \equiv N - N_0$ restantes pertenecen al grupo tratamiento. Por lo tanto, las unidades de control quedan indexadas por $I_0 := \{1, \ldots, N_0\}$, mientras que las unidades tratadas por el conjunto $I_1 := \{N_0 + 1, \ldots, N\}$.

Respecto de las covariables, en todo lo que sigue se asume que su conjunto soporte, denotado $\mathbb{X} \subset \mathbb{R}^K$, es un conjunto convexo y compacto.

Siguiendo a Rubin [1974], supongamos que asociado a cada nivel de tratamiento $w \in \{0,1\}$ existe un "resultado potencial" $Y_i(w)$, y que ambos resultados potenciales se relacionan con el resultado observado de la siguiente manera:

$$Y_i = W_i Y_i(1) + (1 - W_i) Y_i(0), \qquad i = 1, \ldots, N.$$

Si tanto $Y_i(1)$ como $Y_i(0)$ fueran observados, el efecto de tratamiento para la i-ésima unidad (efecto individual) sería sencillamente $Y_i(1) - Y_i(0)$. Sin embargo, para una unidad dada, las variables observadas sólo permiten determinar uno de los resultados potenciales. Esto motiva los siguientes indicadores de efectos *promedio* del tratamiento (ver Imbens and Wooldridge [2009] como referencia general):[1]

$$\tau_{tre} = \mathbb{E}[Y_i(1) - Y_i(0) | W_i = 1] \tag{ATT}$$

$$\tau_{con} = \mathbb{E}[Y_i(1) - Y_i(0) | W_i = 0] \tag{ATC}$$

$$\tau = \mathbb{E}[Y_i(1) - Y_i(0)]. \tag{ATE}$$

[1] En lo que sigue, la probabilidad subyacente se denota $\mathbb{P}$, con valor esperado y varianza denotados por $\mathbb{E}$ y $\mathbb{V}$ respectivamente.

Para cada $w \in \{0,1\}$ y $x \in \mathbb{X}$, definamos las siguientes esperanzas condicionales de la media y la varianza:

$$\mu_w(x) = \mathbb{E}(Y(w) \mid X = x), \qquad \sigma_w^2(x) = \mathbb{V}(Y(w) \mid X = x). \tag{1.1}$$

La función de la media condicional es la función de regresión. Bajo los siguientes supuestos

Supuesto 1. $\exists (Y, X, W) : \; (Y_i, X_i, W_i) \overset{iid}{\sim} (Y, X, W),$ $\square$

Supuesto 2. $\exists\, \delta > 0 : \; \mathbb{P}(W_i = 1 | X_i) \in [\delta, 1 - \delta]$ $a.s,$ $\square$

Supuesto 3. $\{Y_i(1), Y_i(0)\} \perp\!\!\!\perp W_i | X_i$ $a.s,$ $\square$

puede demostrarse que (ver Rosenbaum and Rubin [1983] y Abadie and Imbens [2006]) $\mu_w(x) = \mathbb{E}(Y \mid W = w, X = x)$, a partir de lo cual se tiene que

$$\tau_{tre} = \mathbb{E}\left[Y - \mu_0(X) | W = 1\right] \tag{1.2a}$$

$$\tau_{con} = \mathbb{E}\left[\mu_1(X) - Y | W = 0\right] \tag{1.2b}$$

$$\tau = \tau_{tre}\mathbb{P}(W = 1) + \tau_{con}\mathbb{P}(W = 0) \tag{1.2c}$$

2.2 *Estimación de efecto tratamiento*

Dadas las relaciones en (1.2), la estimación de los efectos tratamiento descansa en la estimación de las funciones de regresión. Si denotamos por $\widehat{\mu}_w(x)$ un estimador de $\mu_w(x)$, los estimadores de los efectos tratamiento en (1.2) son de la forma

$$\widehat{\tau}_{tre} = N_1^{-1} \textstyle\sum_{i \in I_1} \left(Y_i - \widehat{\mu}_0(X_i)\right) \tag{1.3a}$$

$$\widehat{\tau}_{con} = N_0^{-1} \textstyle\sum_{i \in I_0} \left(\widehat{\mu}_1(X_i) - Y_i\right) \tag{1.3b}$$

$$\widehat{\tau} = \frac{N_1}{N}\widehat{\tau}_{tre} + \frac{N_0}{N}\widehat{\tau}_{con} \tag{1.3c}$$

Por ejemplo, los estimadores de *matching* de las funciones de regresión las estiman a través de un promedio ponderado de resultados de unidades del grupo con tratamiento opuesto a aquel de la unidad bajo análisis, usando una ponderación que privilegia a las unidades con covariables más cercanas a aquella bajo análisis. De esta manera, los estimadores *matching* de $\mu_{1-W_i}(X_i)$, que denotamos $\widehat{Y}_i$, tienen la forma:

$$\widehat{Y}_i = \begin{cases} \sum_{j=1}^{N_0} \widehat{\lambda}_{ji} Y_j & \text{si } W_i = 1, \\ \sum_{j=1}^{N_1} \widehat{\lambda}_{ji} Y_{j+N_0} & \text{si } W_i = 0, \end{cases} \tag{1.4}$$

donde los ponderadores $\widehat{\lambda}_{ij}$ deben ser no-negativos y sumar uno. Por ejemplo, el *nearest-neighbor (nn) matching* (Abadie et al. [2004] y Abadie and Imbens [2006]) elige los ponderadores de tal forma que $\widehat{Y}_i$ sea el promedio simple de los resultados de M unidades con los vectores de covariables más cercanos a X_i dentro del grupo $W = 1 - W_i$. Allí, M se fija de manera exógena por parte del investigador.

Notemos que para el caso particular $M = 1$ en lo anterior, dada una unidad tratada i, los ponderadores del *nn-matching* son dados por

$$\left(\widehat{\lambda}_{1i}, \ldots, \widehat{\lambda}_{N_0 i}\right)' = \underset{\lambda \in \Delta_{N_0}}{\operatorname{argmin}} \sum_{j=1}^{N_0} \lambda_j \|X_j - X_i\|_2^2, \tag{$\mathcal{P}_{nn}1$}$$

es decir, $\widehat{\lambda}_{ji}$ es igual a 1 si X_j resulta ser el vector más cercano a X_i entre las covariables de unidades de control, e igual a 0 en caso contrario.

2.3 *Balance*

Como se ha señalado, el sistema (1.1)−(1.3) es válido para cualquier variable de resultado que satisfaga los supuestos 1−3. En extremo, si Y es una función lineal de X, digamos, $Y = \beta' X$, entonces el estimador debería cumplir que (se asume que los errores en la función de regresión son cero en valor esperado)

$$\beta' X_i = \begin{cases} \sum_{j=1}^{N_0} \widehat{\lambda}_{ji} \beta' X_j & \text{si } W_i = 1, \\ \sum_{j=1}^{N_1} \widehat{\lambda}_{ji} \beta' X_{j+N_0} & \text{si } W_i = 0. \end{cases} \tag{1.5}$$

De esta manera, si la unidad i es tratada, el error que se comete con el estimador anterior es igual a $\left|\beta' X_i - \sum_{j=1}^{N_0} \widehat{\lambda}_{ji} \beta' X_j\right|$, el cual, por desigualdad de Cauchy-Schwarz, cumple que

$$\left|\beta' X_i - \sum_{j=1}^{N_0} \widehat{\lambda}_{ji} \beta' X_j\right| \leq \|\beta\| \left\|X_i - \sum_{j=1}^{N_0} \widehat{\lambda}_{ji} X_j\right\|.$$

De anterior, parece natural entonces elegir los ponderadores $\widehat{\lambda}_{ji}$ de modo que dicho error sea minimizado, lo que se aproxima minimizando la discrepancia del lado derecho en la desigualdad anterior. Con dicha elección se logra entonces el "mejor balance posible" del vector X_i en términos de promedios ponderados de covariables de unidades con tratamiento opuesto. Diremos que un vector de ponderadores que hace lo recién indicado es uno que *balancea las covariables de la unidad i* (balance individual).

Sobre lo recién expuesto, notar que, primero, si uno permite que los vectores de ponderación tengan componentes arbitrarias, entonces las covariables de las unidades con tratamiento opuesto a aquella bajo análisis forman una base de $\mathbb{R}^K$ (cosa que ocurre con probabilidad uno bajo los supuestos antes indicados y suponiendo que $K > \max\{N_0, N_1\}$, pues esos vectores están en "posición general"; ver Cover and Efron [1967]).

Segundo, el mejor balance de X_i en términos de promedios ponderados de covariables según se ha indicado queda definido por aquellos vectores de ponderación con los que se construye la proyección de X_i en la envoltura convexa de las covariables de unidades $j \in \{1, \ldots, N\}$, tal que $W_j \neq W_i$.

Tercero, ciertamente puede haber más de un vector de ponderadores que produce tal proyección, es decir, que arrojan el mejor balance posible del vector X_i.

El punto tercero anterior es relevante para este análisis, pues implica que el concepto de balance individual recién expuesto no basta, por sí solo, para definir un estimador de matching. Se requiere algún tipo de refinamiento para elegir entre eventuales combinaciones convexas (ponderadores) que arrojan el mismo balance de covariables. El *blop-matching* propone una forma de implementar dicho refinamiento.

3 *blop-matching*

La descripción del *blop-matching* se hace para una unidad i que recibió el tratamiento (*i.e.*, $i > N_0$). La extensión a unidades de control es directa.

De lo antes expuesto, se tiene que $\widehat{\lambda}_i = (\widehat{\lambda}_{1i}, \ldots, \widehat{\lambda}_{N_0 i})' \in \Delta_{N_0}$ "balanceará óptimamente" el vector de covariables de la unidad i siempre y cuando se cumpla que

$$\sum_{j=1}^{N_0} \widehat{\lambda}_{ji} X_j = \mathrm{Proj}(X_i),$$

donde $\mathrm{Proj}(X_i)$ es la proyección de la covariable X_i en la envoltura convexa de las covariables de unidades con tratamiento opuesto a la unidad i (en este caso, unidades de control), es decir, el único vector que cumple

$$\|X_i - \mathrm{Proj}(X_i)\| = \min_{Z \in \mathrm{co}\left\{X_1,\ldots,X_{N_0}\right\}} \|X_i - Z\| \,.$$

En otras palabras, $\widehat{\lambda}_i \in \Delta_{N_0}$ es solución del siguiente problema de optimización (problema de primer nivel):

$$\min_{(\lambda_{ji})' \in \Delta_{N_0}} \left\| X_i - \sum_{j=1}^{N_0} \lambda_{ji} X_j \right\| . \qquad (\mathcal{P}_0)$$

Reinterpretando el *blop-matching* que proponen Díaz et al. [2015], éste consiste en modificar $\mathcal{P}_{nn1}$ para que los ponderadores también balanceen las covariables de manera individual, lo que logra imponiendo que los vectores que resuelven $\mathcal{P}_0$ definan el conjunto factible de $\mathcal{P}_{nn1}$. De esta manera, el *blop-matching* utiliza los ponderadores que resuelven el siguiente problema de optimización (problema de segundo nivel):

$$\min_{\lambda \in \Delta_{N_0}} \sum_{j=1}^{N_0} \lambda_j \|X_j - X_i\|_2^2 \quad \text{s.a.} \quad \textstyle\sum_{j=1}^{N_0} \lambda_j X_j = \mathrm{Proj}(X_i). \qquad (\mathcal{P}_1)$$

Denotando la solución de $\mathcal{P}_1$ por $\widehat{\lambda}_i^b = (\widehat{\lambda}_{1i}^b, \ldots, \widehat{\lambda}_{N_0 i}^b)' \in \Delta_{N_0}$ en caso que $W_i = 1$, y $\widehat{\lambda}_i^b = (\widehat{\lambda}_{1i}^b, \ldots, \widehat{\lambda}_{N_1 i}^b)' \in \Delta_{N_1}$ cuando $W_i = 0$, se tiene entonces que el estimador *blop-matching* de la función de regresión es

$$\widehat{Y}_i^b = \begin{cases} \sum_{j=1}^{N_0} \widehat{\lambda}_{ji}^b Y_j & \text{si } W_i = 1, \\ \\ \sum_{j=1}^{N_1} \widehat{\lambda}_{ji}^b Y_{j+N_0} & \text{si } W_i = 0, \end{cases}$$

y que los estimadores *blop-matching* de los efectos tratamiento son

$$\widehat{\tau}_{tre}^b = N_1^{-1} \textstyle\sum_{i \in I_1} \left(Y_i - \widehat{Y}_i^b \right) \qquad (1.6)$$

$$\widehat{\tau}_{con}^b = N_0^{-1} \textstyle\sum_{i \in I_0} \left(\widehat{Y}_i^b - Y_i \right) \qquad (1.7)$$

$$\widehat{\tau}^b = \frac{N_1}{N} \widehat{\tau}_{tre}^b + \frac{N_0}{N} \widehat{\tau}_{con}^b. \qquad (1.8)$$

Como se ha expuesto, el ponderador por *blop-matching* es un refinamiento de las soluciones de $\mathcal{P}_0$. Si el problema $\mathcal{P}_0$ tuviese una solución única, resolver $\mathcal{P}_1$ sería innecesario.

Notar que otra solución "refinada" de $\mathcal{P}_0$ a la aquí expuesta se obtendría si uno emplease $\|X_j - X_i\|_2^s$, $s \neq 2$, en la función objetivo del problema $\mathcal{P}_1$. En Díaz et al. [2015] se propone utilizar $s = 1$, mientras que aquí se implementa con $s = 2$. Las soluciones de uno u otro esquema deberían diferir. La elección de $s = 2$ se justifica luego de realizar una expansión de Taylor de segundo orden del sesgo del estimador de efecto tratamiento, mostrando que dicho sesgo se puede acotar superiormente por una combinación lineal (ponderadores positivos) de las funciones objetivo de los problemas $\mathcal{P}_0$ y $\mathcal{P}_1$, donde dichos ponderadores son desconocidos. Para obtener un aproximado al mínimo de dicha cota superior, esos autores proponen que los pesos resuelvan un problema secuencial de optimización, primero minimizando el término que contiene las discrepancias entre covariables y luego, con esas soluciones, minimizando la expresión que contiene discrepancias al cuadrado, lo que da lugar a la implementación que aquí se presenta.

Finalmente, algunos comentarios adicionales sobre el estimador que se ha mostrado:

1. El *blop-matching* exige resolver $\mathcal{P}_0$ y $\mathcal{P}_1$ secuencialmente. En optimización, esto se conoce como un problema de optimización en dos niveles (*blop*, por sus siglas en inglés), de ahí el nombre del método.
2. En la práctica, aquí se resuelve $\mathcal{P}_1$ sobre el conjunto de soluciones del problema $\mathcal{P}_0$, por lo que $\mathcal{P}_1$ es, en definitiva, un problema de programación lineal en forma estándar, con $K+1$ restricciones de igualdad. Dado lo anterior, el número de unidades con tratamiento opuesto a la unidad bajo análisis que tienen ponderación positiva luego de resolver el problema $\mathcal{P}_1$ variará siempre entre 1 y $K+1$. Notar, sin embargo, que para cada unidad bajo análisis ocurre que dichas unidades no necesariamente son sus $K+1$ vecinos más cercanos (en términos de distancia entre covariables), lo que agrega dificultad para resolverlo.

3.1 *Estimación de la Varianza*

Extendiendo las ideas de Abadie and Imbens [2006], Díaz et al. [2015] proponen estimadores consistentes para las varianzas de $\widehat{\tau}$ y $\widehat{\tau}_{tre}$. Estos estimadores no requieren de ningún tipo de simulación, pero sí de una serie de emparejamientos auxiliares.

La estimación de la varianza procede de la siguiente manera:

1. para un individuo indexado por $i \in \{1, \ldots, N_0\}$, se construye una "muestra sintética de opuestos", la que es indexada por $\{1, \ldots, N_0\} \setminus \{i\} = \{1, \ldots, i-1, i+1, \ldots, N_0\}$; para el caso $i \in \{N_0+1, \ldots, N\}$, los índices son $\{N_0+1, \ldots, N\} \setminus \{i\}$,
2. sobre las covariables de las unidades en la muestra sintética de opuestos, y según el caso, se reestima $\mu_{1-W_i}(X_i)$ vía *blop-matching*, guardando el nuevo ponderador como $\widehat{\varphi}_i$, y la nueva imputación como $\widetilde{Y}_i^b$.

Entonces, bajo ciertas condiciones de regularidad (ver Díaz et al. [2015]), los siguientes son estimadores consistentes de la varianza de los estimadores *blop-matching* de los efectos tratamiento:

$$\widehat{V}[\widehat{\tau}^b] = \frac{1}{N^2}\sum_{i\in I}\left\{(Y_i - \widehat{Y}_i^b)^2 + \left[(1+c_i^{[1]})^2 - (1+c_i^{[2]})\right]\widehat{\sigma}_i^2\right\} - \frac{(\widehat{\tau}^b)^2}{N}$$

$$\widehat{V}[\widehat{\tau}^b_{tre}] = \frac{1}{N_1^2}\sum_{i\in I_1}(Y_i - \widehat{Y}_i^b)^2 + \frac{1}{N_1^2}\sum_{i\in I_0}\left((c_i^{[1]})^2 - c_i^{[2]}\right)\widehat{\sigma}_i^2 - \frac{(\widehat{\tau}^b_{tre})^2}{N_1}$$

donde

$$\widehat{\sigma}_i^2 := \frac{(Y_i - \widetilde{Y}_i^b)^2}{1+\|\widehat{\varphi}_i\|_2^2} \qquad c_i^{[\alpha]} := \sum_{j:W_j\neq W_i}\left(\widehat{\lambda}^b_{(i-W_iN_0)\,j}\right)^{\alpha}, \quad \alpha = 1,2.$$

Vea Díaz et al. [2015] para una demostración del resultado.

4 *Implementación*

En la sección anterior, se definió el *blop-matching* como la solución a un problema de optimización en dos niveles. Es fácil ver que dicho problema tiene solución, pero no es obvio cómo llegar a ella. En esta sección, explicaremos brevemente como obtener una solución aproximada asumiendo, sin pérdida de generalidad, que la unidad bajo análisis (i) recibió el tratamiento (*i.e.*, $i > N_0$). La extensión al caso que la unidad bajo es control resulta directa de lo que se expone a continuación.

Para lo que sigue, como referencia general sobre programación lineal, ver el libro de Matousek [2007] el cual, entre otros, introduce algoritmos populares para resolver problemas de programación lineal (PPL). Por otro lado, el libro de Pan [2014] explica cómo implementar dichos algoritmos de manera eficiente. Son las referencias de cabecera que aquí se emplean para implementar el estimador.

4.1 *Implementación del primer nivel*

Para i una unidad tratada, recordemos que $\widehat{\lambda}_i = (\widehat{\lambda}_{1i}, \ldots, \widehat{\lambda}_{N_0 i})' \in \Delta_{N_0}$ resuelve $\mathcal{P}_0$ *ssi*

$$\textstyle\sum_{j=1}^{N_0}\widehat{\lambda}_{ji}X_j = \mathrm{Proj}(X_i) := \underset{Z\in \mathrm{co}\,\{X_1,\ldots,X_{N_0}\}}{\mathrm{argmin}} \|X_i - Z\|\,.$$

Desde el punto de vista de la teoría asintótica, la elección de la norma es irrelevante. Sin embargo, desde un punto de vista computacional, resulta importante. En particular, si uno emplea la norma-2 (norma Euclidiana), el problema de primer nivel $\mathcal{P}_0$ es uno de programación cuadrática, mientras que si uno emplea la norma-1 obtendremos un PPL. Así, por simplicidad para la implementación, en lo que sigue utilizaremos la norma-1 ($\|\cdot\|_1$) en la definición del problema de primer nivel, con el cual se determina la proyección de cierta covariable en la envoltura convexa de las covariables de unidades que tienen tratamiento opuesto.

Denotando $\widehat{c}_i = X_i - \text{Proj}(X_i)$, entonces es claro que

$$(\widehat{c}_i, \widehat{\lambda}_i) \in \underset{(c,\lambda)\in\mathbb{R}^K\times\Delta_{N_0}}{\text{argmin}} \left\{ \|c\|_1 \quad s.a. \quad c + \textstyle\sum_{j=1}^{N_0} X_j\lambda_j = X_i \right\}.$$

Si $c = (c_1, \ldots, c_K)' \in \mathbb{R}^K$, denotemos $|c| = (|c_1|, \ldots, |c_K|)' \in \mathbb{R}^K_+$, y definamos $d = |c|$. Entonces, es claro que $\|c\|_1 = \boldsymbol{\imath}'_K d$. Más aún, usando el preorden parcial en $\mathbb{R}^K$, por la definición de d y $|c|$ existirán dos vectores $\mathbf{0}_K \leq \alpha$ y $\mathbf{0}_K \leq \beta$ tales que

$$\left.\begin{array}{rcl} c &=& d - 2\beta \\ c &=& -d + 2\alpha \end{array}\right\} \Rightarrow \left\{\begin{array}{rcl} c &=& \alpha - \beta \\ d &=& \alpha + \beta \end{array}\right.$$

lo cual implica que $\|c\|_1 = \boldsymbol{\imath}'_K (\alpha + \beta)$.

Combinando todo lo anterior, se tiene que $\text{Proj}(X_i) = X_i - (\widehat{\alpha}_i - \widehat{\beta}_i)$, donde $(\widehat{\alpha}_i, \widehat{\beta}_i, \widehat{\lambda}_i)$ resuelve el siguiente problema de optimización:

$$\min_{(\alpha,\beta,\lambda)\in\mathbb{R}^K_+\times\mathbb{R}^K_+\times\mathbb{R}^{N_0}_+} (\boldsymbol{\imath}'_K\alpha + \boldsymbol{\imath}'_K\beta + \mathbf{0}'_{N_0}\lambda) \tag{$\mathcal{P}^\star_0$}$$

$$s.a. \quad \begin{bmatrix} \boldsymbol{I}_K & -\boldsymbol{I}_K & X_1 & \ldots & X_{N_0} \\ \mathbf{0}'_K & \mathbf{0}'_K & 1 & \ldots & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \lambda \end{bmatrix} = \begin{bmatrix} X_i \\ 1 \end{bmatrix}.$$

Un aspecto destacable del problema auxiliar $\mathcal{P}^\star_0$ es que posee una base factible fácil de calcular. Específicamente,

$$\mathcal{B} \equiv \big\{ \{k : X_{ki} \geq X_{kN_0}\}, \{K + k : X_{ki} < X_{kN_0}\}, 2K + N_0 \big\}$$

define los indices de vectores que conforman la base factible. En efecto, para probar que este conjunto efectivamente marca a $K+1$ columnas linealmente independientes (*l.i.*) dentro de la matriz del lado izquierdo del problema $\mathcal{P}^\star_0$, en primer lugar notar que la columna $2K + N_0$ siempre es *l.i.* de las primeras $2K$, pues es la única que no tiene un cero en su última fila, mientras que las columnas $\{k : X_{ki} \geq X_{kN_0}\} \cup \{K + k : X_{ki} < X_{kN_0}\}$ son siempre ortogonales ya que cada una contiene (salvo por el signo) una columna distinta de $\boldsymbol{I}_{K+1}$.

Para probar que la base anterior es factible, basta con demostrar que la solución básica inducida por $\mathcal{B}$ es no-negativa. Para este caso en particular, dicha solución está dada por $\widetilde{\alpha} = (\widetilde{\alpha}_j)' \in \mathbb{R}^K$, $\widetilde{\beta} = (\widetilde{\beta}_j)' \in$

$\mathbb{R}^K$ y $\widetilde{\lambda} \in \mathbb{R}^{N_0}$ tales que

$$\begin{aligned}
\widetilde{\alpha}_j &= +\,\text{máx}\{X_{ji} - X_{jN_0}, 0\},\\
\widetilde{\beta}_j &= -\,\text{mín}\{X_{ji} - X_{jN_0}, 0\},\\
\widetilde{\lambda} &= \boldsymbol{e}_{N_0 N_0},
\end{aligned}$$

la que es claramente no negativa.

Contando con una base factible inicial $\mathcal{B}$, el problema $\mathcal{P}_0^\star$ puede resolverse mediante el método símplex revisado.

4.2 *Implementación del segundo nivel*

Recordemos que $\widehat{\lambda}_i^b = (\widehat{\lambda}_{1i}^b, \ldots, \widehat{\lambda}_{N_0 i}^b)' \in \Delta_{N_0}$ resuelve $\mathcal{P}_1$, es decir, es solución del problema

$$\min_{\lambda \in \Delta_{N_0}} \textstyle\sum_{j=1}^{N_0} \lambda_j \|X_j - X_i\|_2^2 \quad \text{s.a.} \quad \textstyle\sum_{j=1}^{N_0} \lambda_j X_j = \text{Proj}(X_i)$$

o, equivalentemente, $\widehat{\lambda}_i^b$ es solución de

$$\begin{aligned}
&\min_{\lambda \in \mathbb{R}^{N_0}} \quad \textstyle\sum_{j=1}^{N_0} \lambda_j \|X_j - X_i\|_2^2 && (\mathcal{P}_1^\star)\\
&s.a. \quad \begin{bmatrix} \text{Proj}(X_i) \\ 1 \end{bmatrix} = \begin{bmatrix} X_1 & \ldots & X_{N_0} \\ 1 & \ldots & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_{N_0} \end{bmatrix}\\
&\qquad \mathbf{0} \leq \lambda.
\end{aligned}$$

Comparando $\mathcal{P}_0^\star$ con $\mathcal{P}_1^\star$, se puede apreciar que el vector $\widehat{\lambda}_i$ (parte de la solución del primer nivel) siempre será una solución factible, aunque no necesariamente básica. Dada esta solución, defina

$$\mathcal{C} \equiv \left\{ j : \widehat{\lambda}_{ji} > 0 \right\}.$$

Por construcción, este conjunto identificará a #$\mathcal{C}$ columnas *l.i.* de la matriz de coeficientes. Asumiendo que esta última matriz tenga rango $K+1$, el conjunto $\mathcal{C}$ podrá ser extendido a una base $\mathcal{B}$ siguiendo el método convencional[2]. Más aún, dicha base será factible, pues su solución básica asociada, $\widetilde{\lambda}_i = (\widetilde{\lambda}_{1i}, \ldots, \widetilde{\lambda}_{N_0 i})'$, siempre será no-negativa. Específicamente,

[2] *e.g.* Axler [2014, teorema 2.33].

$$\widetilde{\lambda}_{ji} = \begin{cases} \widehat{\lambda}_{ji} & \text{si } j \in \mathcal{C}, \\ 0 & \text{si } j \notin \mathcal{C}. \end{cases}$$

Una vez obtenida la base factible inicial, el problema $\mathcal{P}_1^\star$ puede resolverse aplicando nuevamente el método símplex revisado.

5 El comando blopmatch

5.1 Descripción

`blopmatch` estima efectos promedio de tratamientos binarios (ATE y ATT) a partir de datos observacionales usando *blop-matching*. Esta variante de *matching* imputa el resultado potencial faltante de cada individuo usando un promedio ponderado de los resultados de los individuos que reciben el otro nivel de tratamiento. El vector de ponderadores para cada sujeto es determinado resolviendo un problema de optimización en dos niveles.

5.2 Instalación

Desde Stata, tipee

```
. net from "https://rawgit.com/igutierrezm/blopmatch/master"
. net install blopmatch, replace  // install blotmatch
. net get blopmatch               // download ancilliary data
```

5.3 Sintaxis

```
blopmatch [if] [in] , ///
   outcome(varname) treatment(varname) covariates(varlist) [options]
```

Argumentos obligatorios

outcome	Especifica el resultado.
treatment	Especifica el tratamiento.
covariates	Especifica la(s) covariables(s).

Argumentos opcionales

Niveles de tratamiento	
ttlevel	Nivel de tratamiento de los tratados. *Default*: 1.
utlevel	Nivel de tratamiento de los controles. *Default*: 0.
Reporte	
level	Fija el nivel de confianza. El *default* es 95 %.
dmvariables	Muestra los nombres de las covariables.
Símplex Revisado*	
otol	Tolerancia del solver (test de optimalidad). *Default*: 1e-8.
btol	Tolerancia del solver (test de acotamiento). *Default*: 1e-8.
imax	Número máximo de iteraciones. *Default*: 1e+3.

*Vea Ferris et al. [2008] o Matousek [2007] para más detalles sobre estos parámetros.

5.4 Detalles

otol. Sea (P) un programa lineal en forma estándar $-$a resolver mediante el método símplex (revisado)$-$ y sea c el actual costo reducido. Es bien sabido que si $c > 0$ la solución actual es óptima. Sin embargo, esta regla es impráctica debido a errores de redondeo. otol relaja esta condición a $c + \text{otol} > 0$.

btol. Sea (P) un programa lineal en forma estándar $-$a resolver mediante el método símplex (revisado)$-$ y sea d la actual columna pivote. Es bien sabido que si $d < 0$ el problema no está acotado. Sin embargo, esta regla es impráctica debido a errores de redondeo. btol relaja esta condición a $d < \text{btol}$.

6 Ejemplo

El archivo `nsw.dta` contiene los datos usados por LaLonde [1986] en su evaluación experimental de la *National Supported Work Demonstration* (en adelante, NSW), un programa de subsidio al trabajo desarrollado en EE.UU. a mediados de la década de los '70. El Cuadro 1.1 describe las variables contenidas en esta base de datos.

El siguiente código carga la base e identifica las variables de respuesta (Y), tratamiento (W) y control (X). Por fines expositivos, crearemos una macro para cada una de estas variables:

```
. use "nsw.dta", clear
. egen stdre75 = std(re75)
. local X age education black hispanic married nodegree stdre75
. local W treat
. local Y re78
```

Variable	*Descripción*
treat	Treatment indicator (1 if treated, 0 otherwise)
age	Age
education	Education
black	1 if black, {space 5} 0 otherwise
hispanic	1 if hispanic, 0 otherwise
married	1 if married, 0 otherwise
nodegree	1 if no degree, 0 otherwise
re75	Earnings in 1975 (in thousands of 1978 dollars)
re78	Earnings in 1978 (in thousands of 1978 dollars)

Cuadro 1.1: Descripción de las variables contenidas en `nsw.dta`.

Una de las virtudes de los datos de LaLonde [1986] es que provienen de un experimento aleatorio, de modo que incluso una diferencia de medias es un estimador insesgado de τ. En particular, el comando

```
. ttest `Y´, by(`W´)
  (output omitted)
```

arroja que el ATE es aproximadamente 886,3. En virtud de lo anterior, cabría esperar que el *nn-matching* arrojara un $\widehat{\tau}$ cercano a 886,3 sin importar el valor de M, pero resulta ser que M afecta $-$a veces, dramáticamente$-$ el resultado. Por ejemplo, los comandos

```
. teffects nnmatch (`Y´ `X´) (`W´), ate metric(euclidean) nneighbor(01)
  (output omitted)
. teffects nnmatch (`Y´ `X´) (`W´), ate metric(euclidean) nneighbor(16)
  (output omitted)
```

arrojan $\widehat{\tau} = 914{,}7$ y $\widehat{\tau} = 1001{,}1$; respectivamente. En cambio (y sin necesidad de fijar ningún parámetro adicional) el comando `blopmatch` arroja $\widehat{\tau} = 936{,}5$:

```
. blopmatch, outcome(`Y´) treatment(`W´) covariates(`X´) dmvariables

Treatment-Effects Estimation
Method                   : Blop Matching
LP Solution Method       : Revised Simplex    No of   treated units =      297
1st Step Norm            : Norm-1             No of untreated units =      425
2nd Step Distance Metric : Euclidean Norm     Solver efficacy       =        1
------------------------------------------------------------------------------
            |     Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
------------+-----------------------------------------------------------------
        ATE |  936.5039   292.5321     3.20   0.001     363.1516    1509.856
        ATT |  610.2753   464.0382     1.32   0.188    -299.2228    1519.773
------------------------------------------------------------------------------

Outcome            : re78
Treatment          : treat
Covariates         : age education black hispanic married nodegree stdre75
Level of treated   : 1
Level of untreated : 0
```

Bibliografía

A. Abadie and G. W. Imbens. Large sample properties of matching estimators for average treatment effects. *Econometrica*, 74:235–267, 2006.

A. Abadie, D. Drukker, J. Leber Herr, and G. W. Imbens. Implementing matching estimators for average treatment effects in stata. *Stata Journal*, 4:290–311, 2004.

S. Axler. *Linear Algebra Done Right*. Springer, 3 edition, 2014.

T. M. Cover and B. Efron. Geometrical probability and random points on a hypersphere. *The Annals of Mathematical Statistics*, 38(1):213–220, 1967.

J. Díaz, T. Rau, and J. Rivera. A matching estimator based on a bilevel optimization problem. *The Review of Economics and Statistics*, 97:803–812, October 2015.

M. Ferris, O. Mangasarian, and S. Wright. *Linear Programming with MATLAB (MPS-SIAM Series on Optimization)*. Society for Industrial and Applied Mathematics, 2008.

G. W. Imbens and J. M. Wooldridge. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature*, 47 (1):5–86, March 2009.

R. LaLonde. Evaluating the econometric evaluations of training programs. *American Economic Review*, 76:604–620, 1986.

J. Matousek. *Understanding and Using Linear Programming*. Springer, 2007.

P. Q. Pan. *Linear Programming Computation*. Springer, 2014.

P. R. Rosenbaum and D. B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70:41–55, 1983.

D. B. Rubin. Estimating causal effects of treatments in randomised and non-randomised studies. *Journal of Educational Psychology*, 66: 688–701, 1974.