A Stata package for the estimation of the dose-response function when the treatment is multidimensional

Enrico Cristofolletti
Guidelines for authors
Papers may be written in Italian or in English. Faculty members of the Department must submit to one of the editors in pdf format. Management papers should be submitted to R. Gabriele. Economics Papers should be submitted to L. Andreozzi. External members should indicate an internal faculty member that acts as a referee of the paper.

Typesetting rules:
1. papers must contain a first page with title, authors with emails and affiliations, abstract, keywords and codes. Page numbering starts from the first page;
2. a template is available upon request from the managing editors.
A Stata package for the estimation of the dose-response function when the treatment is multidimensional

Enrico Cristofoletti

Abstract

Propensity score methods are wildly used techniques for the evaluation of causal effects in observational studies. Although Rosenbaum and Rubin's (1983) original article focused solely on binary treatments, further studies generalize the methods to multi-valued treatments, continuous treatments, and multidimensional continuous treatments. Despite its potential, Stata offers plenty of packages for all the cases but the last one. This paper aims to introduce a new Stata package – GPSMD – that implements the propensity score generalization to multidimensional continuous treatment developed by Egger and von Ehrlich (2013). The article illustrates the econometric framework and presents the commands implemented. We finally go through a simple working example to show the commands and the capability of the method to overcome bias.

Keywords: continuous multiple treatments, GPSMD, dose-response, generalized propensity score

* Department of Economics and Management, University of Trento, Trento (Italy); enrico.cristofoletti@unitn.it
1. Introduction

Since Rosenbaum and Rubin's (1983) groundbreaking article, propensity score (PS) methods have become widely used instruments in the evaluation of causal effects in observational studies (Pearl 2010; King and Nielsen 2019). Unlike in (ideal) randomized experiments – where the treatment is randomly assigned to different groups and thus exposed and unexposed can be considered exchangeable (Hernán and Robins 2006) –, in observational studies, the treatment assignment is not governed by the researcher and, therefore, exchangeability is an issue. The consequence is that in the ideal case of random experiments the effect of exposure can be recovered by simply using association measures, in fact, the groups differ only for the reception of the treatment. In observational studies, the indiscriminate use of association measures gives a misleading estimation of the causal effect because different groups likely have different compositions (Hernán and Robins 2006). PS methods are born to reestablish the balance between exposed and non-exposed groups so that the causal effect is identifiable. In particular, Rosenbaum and Rubin (1983) show that, in settings where balance is undermined by selection on observables (Cerulli 2015), conditioning on the propensity score (i.e. the probability of being treated conditional on observable covariates) succeeds in restoring the balance between groups.

Originally, PS has been developed for binary treatments (Rosenbaum and Rubin 1983). More recent extensions include the generalized propensity score (GPS) for multi-valued treatments (Imbens 2000) as well as continuous treatments (Hirano and Imbens 2004; Imai and Van Dyk 2004). An extension of Hirano and Imbens (2004) to the case in which the treatment is composed of more than one continuous dimension is provided by Egger and von Ehrlich (2013). While Stata makes available a large set of packages for the estimation of propensity score – see Guo and Fraser (2015) and Caliendo and Kopeinig (2008) for an overview mainly devoted to binary treatments, and Bia and Mattei (2008), Guardabascio and Ventura (2014), Bia et al. (2014) for continuous treatments –, none of them enables the analysis adopting the multidimensional framework. This is probably the reason why only a handful of studies (Peter Hannes Egger and Egger 2016; Peter H. Egger and Lassmann 2018; Peter H. Egger, Ehrlich, and Nelson 2020; Erhardt 2017) adopts the multidimensional framework despite its potential usefulness. Many policies are better conceptualized as a multidimensional treatment rather than binary or monodimensional. For example, the effect of financial aids can differ depending on the type of investments that subsidies trigger. Moreover, different policies can be contemporarily adopted and their effects can be seen only by considering them together.

The aim of this paper is, then, to present a Stata package for the estimation of dose-response function in presence of multidimensional treatment when the dimensions are continuous. The package has been named GPSMD. The structure of the paper is the following: first, we will describe Egger and von Ehrlich’s econometric framework; second, we will describe the different commands; third, we will present a simple example by using simulated data in order to show the commands and the performance of the method. The last section concludes and presents some further modifications that we mean to implement.
2. The econometric outline

In this section is summarized the econometric framework defined by Egger and von Ehrlich (2013). The actual treatment level experienced by the $i^{th}$ observational unit is a random vector $T_i = (T_{1i}, ..., T_{Mi})'$ where $M$ is the number of dimensions of the treatment. There are $N$ observational units in the sample.

The outcome $Y_i$ is assumed to be a function of the treatment $Y_i(T_i) = h(T_{1i}, ..., T_{Mi})$. The potential outcome is defined as: $Y_i(t), t \in \mathcal{S}$. Where $\mathcal{S}$ is the set of all potential treatments. The average dose-response function is defined as:

$$
\mu(t) \equiv E[Y_i(t)]
$$

The level of the treatment is defined by an $m$-equation structural model. The reduced equations are defined as:

$$
T_{mi} = f(Z_i^*, y_m) + v_{mi}, m = 1, ..., M
$$

Where $Z_i$ is a vector formed by the union of the exogenous variables $X_{mi}$ and possibly their interactive terms\(^1\).

For the identification, weak unconfoundedness is assumed, that is:

$$
Y_i(t) \perp T_i | Z_i \forall t \in \mathcal{S}
$$

This means that the potential outcome at level $t$ of treatment is independent of the actual treatment status $T_i$ when we condition on the covariates $Z_i$ and that this is true for all treatments.\(^2\) Simply controlling for covariates can induce a problem concerning the dimensionality of the model: the solution is conditioning on the propensity score.

The density of observing the treatment $T_i = t = (t_{1i}, ..., t_{Mi})'$ conditional on the exogenous variables

\(^1\) The choice of using the same set of variable in estimating the reduced equation can be scarcely parsimonious. We are planning to implement also the possibility of using different sets of covariates in the reduced equations. This will require the implementation of the SUR estimator in order to increase efficiency (Cameron and Trivedi 2009). We thanks Giovanni Cerulli for the suggestion.

\(^2\) This imply that we can estimate the average treatment effect by using those that actually have received the treatment: $E[Y_i(t)|Z_i] = E[Y_i(t)|T_i = t', Z_i] = E[Y_i(t)|T_i = t, Z_i] = E[Y_i|T_i = t, Z_i]$. (Imbens 2000)
\[ g(t, z) \equiv f_{T_i|Z_i}(T_i = t|Z_i = z) \]

Then the propensity score is defined as \( G_i = g(T_i, Z_i) \). It is worth noting that the generalized propensity score defines the random variable \( G_i = g(T_i, Z_i) \), as a transformation of both \( T_i \) and \( Z_i \), as well as the family of random variables indexed by \( t \), \( g(t, Z_i) \) (Imbens 2000).

By construction, the generalized propensity score implies the balance property. Loosely speaking:

\[ Z_i \perp 1\{T_i = t\}|g(t, Z_i) \forall t \in \mathfrak{Z} \]

This means that, once controlled for \( g(t, Z_i), Z_i \) and the treatment are independent (see Appendix 1 for the proof).

As a consequence, under weak unconfoundedness, it can be shown (Peter H. Egger and von Ehrlich 2013) (see Theorem 1 in Appendix 1) that, once conditioned on the propensity score, the potential outcome is independent of the treatment:

\[ Y_i(t) \perp T_i|g(t, Z_i), \forall t \in \mathfrak{Z} \]

And, thus, that conditioning on \( g(t, Z_i) \) is like if we conditioned on covariates\(^3\).

This implies the following (see Theorem 2 in Egger and von Ehrlich (2013)):

\[ E[Y_i|T_i = t, g(T_i, Z_i)] = E[Y_i(t)|T_i = t, g(t, Z_i)] = E[Y_i(t)|g(t, Z_i)] \]

And, then, that by using the law of iterated expectation the average dose-response function can be retrieved:

\[ \mu(t) \equiv E[Y_i(t)] = E[E[Y_i(t)|g(t, Z_i)]] \]

\(^3\) It is worth noting that independence, as well as the balancing property, hold within strata of propensity score calculated at a given treatment \( t \), \( g(t, Z_i) \), not within strata of the propensity score \( g(T_i, Z_i) \).
The practical implementation consists in estimating $E[Y_i | T_i = t, g(T_i, Z_i)]$ by a flexible polynomial function with $g(T_i, Z_i)$ as a covariate. Then we average over the propensity score for retrieving the dose-response function. If the flexible polynomial function is correctly specified, then GPSMD should reduce the bias of the estimation.

Summarizing, as in the case of mono-dimensional continuous treatment, the steps involved in the implementation of the GPS method are:

1. Estimating the Generalized propensity score (command: gpsMD)
2. Enforcing the common support if needed (command: CommSupp)
3. Testing the balancing property (command: gpsMDbal)
4. Estimating the dose-response function (command: gpsMDPolEs)

In the following, we describe the commands and algorithms for implementing each step. Every section about the presentation of a command will include the description of the algorithm, the syntax of the program, and the description of the objects and e-class objects generated by the program. The printed output will be described in the section with the application.

### 3. gpsMD

The first step in the application of the GPS method is the estimation of the propensity score. We provide the command: gpsMD. In the following, the algorithm is described as well as the command syntax.

Recalling that the reduced equations are defined as:

$$T_{mi} = f(Z_i, \gamma_m) + v_{mi}, m = 1, ..., M$$

Where $Z_i$ is the vector including the union of the exogenous variables $X_{mi}$ and possibly their interactive terms. In the implementation of gpsMD, we assume errors having a multivariate normal distribution:

$$v_i = (v_{1i}, ..., v_{Mi})' \sim \mathcal{N}(0_m, \Sigma)$$

The assumption on the errors implies that $T_i | Z_i$ is distributed: $T_i | Z_i \sim \mathcal{N}(f(Z_i, \gamma_m), \Sigma)$.

In order to estimate the propensity score we proceed in 4 steps:

1. The $m$ reduced equations are estimated by OLS;
2. The $m$ vectors of residuals are predicted;
3. By using the residuals, the variance-covariance matrix, $\Sigma$, is estimated: $\Sigma = \text{Cov}(v_1, ..., v_M)$ where $v_m = [v_{1m}, ..., v_{Nm}]$;
4. Finally, the generalized propensity score for each observational unit is estimated by:
\[
G_i = \frac{1}{(2\pi)^{\frac{M}{2}} \det(\Sigma)^{\frac{1}{2}}} \exp \{- \frac{1}{2} \mathbf{v}' \Sigma^{-1} \mathbf{v}_i \}
\]

This implies the following formula for the vector \( \mathbf{G} = (G_1, ..., G_N)' \):

\[
\mathbf{G} = \frac{1}{(2\pi)^{\frac{M}{2}} \det(\Sigma)^{\frac{1}{2}}} \exp \{- \frac{1}{2} \text{diagonal}(\mathbf{v}' \Sigma^{-1} \mathbf{v}) \}
\]

Where \( \exp(.) \) is now a function for the elementwise exponentiation of a matrix, \( \mathbf{v}' = \begin{bmatrix} v_1' \\ \vdots \\ v_N' \end{bmatrix} \), and \( N \) is the number of observational units in the sample.

If the treatments are deemed to follow a multivariate log-normal distribution, the program calculates (option \texttt{ln()} ) the conditional density using the log transformation of the dimensions of the treatment in the list. Then, in order to recover the propensity score of the untransformed treatment, the conditional density calculated on the transformation is divided by the dimensions of the treatment that have been transformed (Dobrow 2013, para. 6.6).

To select the model the Akaike's Information Criterion (Cavanaugh and Neath 2019) could be used. The command \texttt{AkaikeMax}, which identifies the model which minimizes the Akaike's Information Criterion is described in Appendix 2.

\textbf{Syntax (gpsMD)}

\begin{verbatim}
gpsMD varlist(min=1) , exogenous(varlist) gpsMD(string)  
[chosenpoint(string) ln(varlist)]
\end{verbatim}

\begin{itemize}
  \item \texttt{varlist(min=1)} : the dimensions of the treatment.
  \item \texttt{exogenous(varlist)}: the list of the exogenous variable, as well as their possible interactions and powers, depending on the model the user has in mind.
  \item \texttt{gpsMD(string)}: the name for the variable where the generated propensity score will be stored.
  \item \texttt{chosenpoint(string)}: the name of the Stata column vector with the point at which we want to calculate the propensity score (mostly for programs). It is an option that enables the user to generate the propensity score calculated at a given point. It generates \( g(t, Z_i) \) instead of \( g(T_i, Z_i) \).
  \item \texttt{ln(varlist)} : the treatment dimensions that have to be log-transformed.
\end{itemize}

\textbf{Variables generated (gpsMD)}

The variable named as specified in \texttt{gpsMD(string)} with the estimated propensity score.
E-class objects generated (gpsMD)

macros:

e(gpsMDvar) : macro with the string in the option gpsMDvar.

e(Exogenous) : macro with the varlist in the option Exogenous.

e(Dimensions) : macro with the varlist of the dimensions of the treatment.

e(cmdline#) : macro with the cdmline of the reduced equation for dimension #. The user may want to run again only one of the regressions and focus on those results. This macro enables to do it easily.

e(cmd) : macro with the name of the command just invoked (gpsMD).

e(cmdline) : macro with the cdmline. This macro includes the command just invoked including options and specifications.

e(chosenpoint) : macro with the name of the column vector with the chosen point.

e(LNVarCreated) : if the ln(varlist) option is specified, the program generates variables named LN_var consisting in the logarithmic transformation of the variables in the varlist. e(LNVarCreated) contains the list of the variable generated.

e(DimensionsFS) : macro with the name of the dimension used in calculating the propensity score. It differs from e(Dimensions) only if the ln(varlist) option is used.

matrices:

e(VarCov) : the estimated variance-covariance matrix, $\Sigma$.

4. ComSupp

Egger et al. (Peter Hannes Egger and Egger 2016; Peter H. Egger, Ehrlich, and Nelson 2020) generalize to the multidimensional case Flores et al’s (2012) methodology to identify the common support in the case of continuous treatment. The rationale is similar to that of the “minima and maxima comparison” for binary treatments (Caliendo and Kopeinig 2008) which consists in reducing the sample to those observations whose propensity score is higher than the maximum of the minimums of treated and controls group, as well as lower than the minimum of the maximums of treated and controls group. The main difference is that continuous dimensions must be discretized in order to apply a similar criterion: we do not have “treated” and “not treated” anymore. Moreover, differently to the binary case, for each discrete
treatment, it must be chosen a treatment point that represents the discrete set and at which the GPS will be calculated\(^4\). The outline of the algorithm follows.

We partition each dimension of the treatment \(T_{m_1}, m = 1, \ldots, M \) in \(L\) sets such that:

\[
\text{range}(T_{m_1}) = \bigcup_{l=1}^{L} T_{m_1}^l, m = 1, \ldots, M
\]

\(T\) is then discretized in \(M \times L\) sets defined by \(\times_{m=1}^{M} \{T_{m_1}^1, \ldots, T_{m_1}^L\}_m\). Denote this set as \(J\).

Then for each discrete subset of the treatment \(T^D \in J\):

1. we chose a representative point (e.g. mean, median) \(\bar{\mathbf{t}}^D\);
2. we evaluate the GPS at that point \(g_j(\bar{\mathbf{t}}^D, \mathbf{Z}_i)\), for each observation in the sample;
3. we keep only those observations such that their GPS calculated at point 2 satisfies:

\[
g_j(\bar{\mathbf{t}}^D, \mathbf{Z}_i) \in \left[ \min_{j \in T^D} \{g_j(\bar{\mathbf{t}}^D, \mathbf{Z}_j)\}, \min_{j \in T^D} \{g_j(\bar{\mathbf{t}}^D, \mathbf{Z}_j)\}, \min_{j \in T^D} \{g_j(\bar{\mathbf{t}}^D, \mathbf{Z}_j)\}, \min_{j \in T^D} \{g_j(\bar{\mathbf{t}}^D, \mathbf{Z}_j)\} \right].
\]

The command does not eliminate observations. It only generates a variable so that the user is free to inspect the characteristics of observations inside and outside the common support.

**Syntax (ComSupp)**

\[
\text{ComSupp} \text{ varlist(min=1)} , \text{ exogenous(varlist) index(string) cutpoints(numlist integer max=1) obs_notsup(string) [testing(numlist integer max=1) ln(varlist)]}
\]

**varlist(min=1)**: the treatment dimensions in the same order as the gpsMD command.

**exogenous(varlist)**: exogenous variables in the same order that in the gpsMD command.

**index(string)**: the point \(\bar{\mathbf{t}}^D\) where the user wants to calculate the GPS. It can be "mean" or "p50": "mean" for the mean, and "p50" for the median.

**cutpoints(numlist integer max=1)**: the number of discrete intervals of the dimensions of the treatment.

\(^4\) As underscored above, an analogy exists between the propensity score for binary treatment and the GPS calculated at a given treatment point, not between the propensity score for binary treatment and the GPS as such.
obs_notsup(string): the name for the dummy variable that takes value 1 if the observation is outside the common support and 0 if the observation is inside the common support.

testing(numlist integer max=1): the user may want to inspect the distribution of the GPS calculated at the representative point of the discrete subsets of the treatment, $g_i(t_{rp}, Z_i)$. If testing is set to 1, the program generates one variable for each discrete subset of the treatment storing, for all observations, the GPS calculated at the representative point of that discrete subset of the treatment. These variables are named obs_notsup# where obs_notsup is the name specified in obs_notsup(string) and # stands for the number of the discrete subset. The dummy variable with the observation indicating if the observation is inside the common support is named simply as in obs_notsup(string).

ln(varlist): the treatment dimensions that have to be log-transformed.

Variables generated (ComSupp)

As explained above, if testing is different from 1, the program generates a variable named as in obs_notsup which takes value 1 if the observation is outside the common support and value 0 if the observation is within the common support.

Note that ComSupp is a nclass command, so you can issue it after having issued gpsMD and before gpsMDBal without incurring an error.

5. gpsMDbal

The gpsMDbal command tests if the balancing property of the GPS holds. A strategy similar to Bia e Mattei is implemented (Bia and Mattei 2012; Bia and Mattei 2008; Peter H Egger and Erhardt 2014; Hirano and Imbens 2004; Guo and Fraser 2015). Recalling that the balancing property is loosely defined as:

$$Z_i \perp 1\{T_i = t\}|g(t, Z_i), \forall t \in \mathcal{S}$$

The command tests whether conditioning on the propensity score is effective in removing the differences in respect to the mean of the exogenous covariates between groups treated with different doses. The procedure is the multidimensional analog to the t-tests for equality of means before and after matching.

---

5 Although the balancing property is a statement about distribution, as customary, the implementation focus only on the first moment (Lee 2013).
implemented in \texttt{pptest} for binary treatments (Leuven and Sianesi 2003). The test consists of the following steps:

1. We partition each dimension of the treatment $T_{ml}$, $m = 1, \ldots, M$ in $L$ sets such that:

   $$\text{range}(T_{ml}) = \bigcup_{l}^{L} T_{ml} \ m = 1, \ldots, M$$

   $T$ is then discretized in $M \times L$ sets defined by $\times_{m=1}^{M} \{T_{ml}^1, \ldots, T_{ml}^L\}_m$. Denote this set as $J$ and a single discrete treatment as $T^D$.

2. As customary, at first, for each exogenous covariate $Z$ and for each $T^D$, independence is assessed before conditioning on propensity score: That is we check whether:

   $$\tilde{Z}_{T^D} = \tilde{Z}_{T^D'} \ T^D \neq T^D' \ T^D, T^D' \in J$$

   Or

   $$\tilde{Z}_{T^D} = \tilde{Z}_{T^-D} \ T^D \in J; T^-D = \bigcup J \setminus T^D$$

Where $\tilde{Z}_{T^D}$ is the sample mean for a given exogenous covariate calculated using the observations in $T^D$, $\tilde{Z}_{T^D'}$ is the sample mean for a given exogenous covariate calculated using the observations in a discrete set different from $T^D$, and $\tilde{Z}_{T^-D}$ is the sample mean for a given exogenous covariate calculated using the observations in the union of the discrete sets different from $T^D$.

(In the package it is implemented the second test through the command \texttt{ttest} for unpaired two-sample).

3. Then, the program tests whether differences remain if we condition on the propensity score. Therefore, it stratifies the sample by propensity score and estimates the differences in the mean between groups with different treatments, within homogeneous propensity score strata. Following Bia (2008), the following algorithm is iterated over each discrete set of the treatment $T^D$:

   a. A representative point $\tilde{i}_{T^D}$ is chosen (e.g. mean, median);
   b. The generalized propensity score $g_i(\tilde{i}_{T^D}, Z_i)$ is calculated for each observational unit.
   c. The propensity score is then discretized in a finite number of intervals. Denote a single discrete interval as $g(\tilde{i}_{T^D}, Z_i)^D$.
   d. Then for each variable in the propensity score, the program tests if the following average is different from 0.

   $$\frac{1}{N} \sum_{g(\tilde{i}_{T^D}, Z_i)^D} N_{g(\tilde{i}_{T^D}, Z_i)^D} \left( \tilde{Z}_{T^D} g(\tilde{i}_{T^D}, Z_i)^D - \tilde{Z}_{T^-D} g(\tilde{i}_{T^D}, Z_i)^D \right)$$

   Where the sum is over the intervals of the propensity score, $N$ is the number of observational units in the sample, and $N_{g(\tilde{i}_{T^D}, Z_i)^D}$ is the number of observations in a given interval of the propensity score. $\tilde{Z}_{T^D} g(\tilde{i}_{T^D}, Z_i)^D$ and $\tilde{Z}_{T^-D} g(\tilde{i}_{T^D}, Z_i)^D$ are the sample mean of the exogenous variable for those observations that belong to the set $T^D \cap g(\tilde{i}_{T^D}, Z_i)^D$ and $T^-D \cap g(\tilde{i}_{T^D}, Z_i)^D$ respectively (in Appendix C the test statistic is derived).
**Syntax (gpsMDbal)**

```plaintext
gpsMDbal varlist (min=1), cutpoints(numlist max=1) index(string) nq_gpsMD(numlist max=1) discrTreat(string) [ptile(string) obs_notsup(string) gpsMDTequalt(string) ln(varlist) ]
```

**varlist (min=1)**: The variables for which the balancing property has to be assessed.

**cutpoints (numlist max=1)**: the number of discrete intervals of the dimensions of the treatment (min 2).

**index (string)**: the point $\bar{\mathbf{t}}_{T,D}$ where the user wants to calculate the GPS. It can be "mean" or "p50": "mean" for the mean, and "p50" for the median.

**nq_gpsMD (numlist max=1)**: the number of discrete subsets of the GPS.

**discrTreat (string)**: the program discretizes the treatment in a user-defined number of subsets. It also generates a variable with the information about the discrete subset an observation belongs to. In discrTreat (string) the user must specify the name of this variable.

**obs_notsup (string)**: the string with the name of the dummy variable generated by the command ComSupp and indicating for each observation its inclusion in the common support. If specified, ComSupp must have been run before issuing gpsMDbal. If it is not specified, gpsMDbal will perform the analysis using the entire sample.

**ptile (string)**: the discrete subsets of the treatment are generated by the cartesian product of the discrete intervals of the dimensions. The program generates a variable for each dimension with the information about the discrete subset of the dimension an observation belongs to. In ptile (string) the user must specify the incipit for the name of these variables.

**gpsMDTequalt (string)**: the user may want to inspect the distribution of the GPS calculated at the representative point of the discrete subsets of the treatment, $g_i(\bar{\mathbf{t}}_{T,D}, \mathbf{Z}_i)$. If gpsMDTequalt (string) is specified, the program generates one variable for each discrete subset of the treatment storing, for all observations, the GPS calculated at the representative point of that discrete subset of the treatment. These variables are named gpsMDTequalt# where gpsMDTequalt is the name specified in gpsMDTequalt (string) and # stands for the number of the discrete subset. By default, the program does not generate these variables.

**ln (varlist)**: the treatment dimensions that have to be log-transformed.

**level (numlist max=1)**: the program prints the table with the adjusted and unadjusted differences in means (e(NofDiscTreat)) both entirely and setting to missing cells whose p-value is higher than a certain threshold (specified in level()). The default is 0.05.
Variables generated (gpsMDbal)

One variable named as specified in `discrTreat`. It is reported, for each observation, the discrete set of the treatment to which the observation is assigned.

One variable named `ptile'# for every dimension # of the treatment. For each observation is reported the discrete set of the #dimension to which the observation is assigned. `discrTreat` is generated as the Cartesian product of `ptile'#.

E-class object generated
Some objects are simply copied from `gpsMD` results since the user may find to have them also after having run `gpsMDbal` useful.

Macros:

- `e(NofDiscTreat)`: a macro with the number of discrete treatments.
- `e(cmd)`: a macro with the command.
- `e(cmdline)`: a macro with cdmline. This macro consists of the command just invoked and includes options and specifications.
- `e(DimensionsFS)`: macro with the name of the dimension used in calculating the propensity score. It differs from `e(Dimensions)` only if the `ln(varlist)` option is used.
- `e(LNVarCreated)`: if the `ln(varlist)` option is used, `e(LNVarCreated)` contains the list of the variables generated by `gpsMD`.

Matrices:

- `e(TabellaImpRes)`: matrix having one row for each variable the user wanted to test and two columns for each discrete treatment. In the cells, the p-value of the test before and after adjusting for the GPS is reported.
- `e(ResultAdj#)`: for each discrete subset of the treatment the program generates a matrix reporting the results of the t-test for the adjusted mean. The first column reports the t statistic, the second column reports the p-value, and the third reports the degree of freedom. There is one row for each variable that the user wanted to test.
- `e(Result#)`: the program generates a matrix reporting the results of the t-test for discrete subset # before the adjustment. There is one column for every r-class object of `ttest` plus one for the estimated difference. There is one row for each variable that the user wanted to test. `e(ResultAdj#)` and `e(Result#)` are somehow redundant objects. The important information is already included in `e(TabellaImpRes)`. 
The program reports, for each discrete subset of the treatment #, a matrix storing the representative treatment vector $\tilde{T}_D$ chosen.

6. gpsMDPolEst

This command implements the estimation of the outcome given the treatment and the propensity score. The command estimate models of the form:

$$E[Y_i|T_i, G_i(T_i, Z_i)] = \alpha_0 + \sum_{m=1}^{M} \sum_{j=1}^{k} (\alpha_{m} T_{mi}^j + \alpha_{G_{ij}} G_{i}^j + \alpha_{mg_{ij}} (G_{i} \cdot T_{i})^j + \alpha_{m \ln G_{i}} (\ln G_{i})^j + \alpha_{m \ln G_{i}} (\ln G_{i} \cdot T_{i})^j)$$

Recalling that $E[Y_i|T_i, G_i(T_i, Z_i)] = E[Y_i|T_i = t, G_i(T_i = t, Z_i)] = E[Y_i|g(t, Z_i)]$. Now the problem is that we must calculate $E[E[Y_i|g(t, Z_i)]]$. Since we do not observe for each $t$ the entire distribution of $g(t, Z_i)$ and that the distribution of $g(t, Z_i)$ — and, thus, of $E[Y_i|g(t, Z_i)]$ — depends entirely on the distribution of $Z_i$, we can use the sample distribution of $Z_i$ to predict $\hat{g}(t, Z_i)$ and, then, we can use the polynomial just estimated to predict $E[Y_i|t, \hat{g}(t, Z_i)]$.

Therefore, the average response function is estimated for a given $t$ as:

$$E[Y(t)] = \frac{1}{N} \sum_{i=1}^{N} E[Y_i|t, \hat{g}(t, Z_i)]$$

In addition to the dose-response function, the program gpsMDPolEst estimates the partial derivatives for each dimension of the treatment.

In estimating confidence intervals, the t-approximation is not reliable because the model includes the propensity score, which is a generated regressor (Wooldridge 2010). Confidence intervals for the response function are thus estimated through bootstrap. Although the bias-corrected accelerated method (BCa) would have been more reliable, it would have been too computationally demanding. Therefore, the program calculates the Bias corrected method (BC) (Carpenter and Bithell 2000; Efron and Tibshirani 1994).

More specifically, for $B$ bootstrap samples, the entire procedure for obtaining the response function and partial derivatives is replicated (starting from the estimation of the reduced equations). The confidence interval at the $1 - \alpha$ level is calculated as:
\[
\left( F_{\theta}^{-1} \left( \Phi \left( 2b + \frac{za}{2} \right) \right), F_{\theta}^{-1} \left( \Phi \left( 2b - \frac{za}{2} \right) \right) \right)
\]

Where \( F_{\theta}^{-1} \) is the inverse of the bootstrap empirical distribution function, \( \Phi(\cdot) \) the cumulative distribution function of the normal distribution, and \( za = \Phi^{-1} \left( \frac{a}{2} \right) \). \( b = \Phi^{-1} \left( \frac{\theta^{*} - \tilde{\theta}}{B} \right) \) is the correction for the bias, and \( \tilde{\theta} \) is the parameter estimated from the original sample.

The quantiles are calculated according to Carpenter and Bithell (2000): \( Q = (B + 1) \Phi \left( 2b \pm \frac{za}{2} \right) \). If needed, the results are rounded and if \( Q > B \), \( Q \) is set to \( B \). Similarly, if \( Q = 0 \), \( Q \) is set to 1.

The program does not produce any graph. Drawing more than two-dimensional graphs in Stata is not an easy task and any graph would require some adjustments. Then, the output of the program consists of several datasets that can be furtherly processed with programs like `graph3d` (Rostam-Afschar and Jessen 2014), `plotmatrix` (Mander 2019; Prã¶g 2019), or `graph twoway contour`.

**Syntax (gpsMDPolEst)**

\[
gpsMDPolEst \text{ outcome treatment dimensions, gpsmd(string) model(string)} \\
\text{ exogenous(varlist) file_pred(string) numboot(numlist integer max=1)} \\
\text{ [dividingint(numlist integer max=1) matrtreat(string) level(numlist max=1) cutpoints(numlist integer max=1) index(string) ln(varlist) matrixwithresults(string)]}
\]

**model(string):** a string with the right side of the model. The right side of the model must be explicitly written due to how the program parses inputs (e.g. "T1 + T2 + gps + T1*gps + T2*gps + T1^2 + T2^2 + (gps^2) + ((T1*gps)^2) + (T2*gps)^2 + ln(gps) + (ln(gps))^2 + (T2*ln(gps))^2 + T2*ln(gps) ")

**dividingint(numlist integer max=1):** if `matrtreat(string)` is not specified the program generates a matrix by dividing the dimensions in `dividingint` number of intervals. The Cartesian product of the extremes of the intervals in the different dimensions constitutes the set of treatment points for which the program estimates the response. The set of treatment points will be stored in `e(matrtreat)` as a matrix with \((\text{dividingint} + 1)\text{number of dimensions}\) rows and columns equal to the number of dimensions.

**matrtreat(string):** The user can specify the treatment points for which she is interested in estimating the response. Treatment points must be stored in a Stata matrix named as specified in `matrtreat(string)`. The matrix must have one column for each treatment dimension so that a single point is identified by a row. Only one option between `dividingint` and `matrtreat` can be specified.
**exogenous(varlist)**: the exogenous variables the user wants to use in the reduced equations.

**file_pred(string)**: As explained above the program does not generate any graphs but only various datasets (see below for a more detailed description of the files generated) with the necessary information for the user to generate the desired graphs. In **file_pred(string)** the user must specify the incipit of the name for the files .dta storing the results.

**level(numlist max=1)**: the confidence level for the confidence intervals (default 0.05).

**numboot(numlist integer max=1)**: the number of bootstrap samples. Since bootstrapping is the only way to obtain the confidence intervals this is not an optional argument.

**cutpoints(numlist integer max=1)**: the number of discrete intervals of the dimensions of the treatment when you calculate the common support (It is worth noticing that when common support is required, the estimation of the dose-response function is done by using only observations that lie on the common support). We suggest using the same number which has been used in calculating the common support.

**index(string)**: the point $\bar{t}_p$ where the user wants to calculate the GPS. It can be "mean" or "p50": "mean" for the mean, and "p50" for the median. **ln(varlist)**: the treatment dimensions that have to be log-transformed.

**matrixwithresults(string)**: if “T” a matrix called e(returnresults) including all the results as well as the chosen doses is returned. Default is “T”. If “F” the matrix is not generated. This option can be useful if the number of treatment points exceeds Stata matrix limits.

*Variables generated (gpsMDPolEst)*

The command generates the variables specified in **model** but the treatment dimensions and the GPS. All the variables are named starting with **I_**, **P_**, or **LN_**. **I_** represent interaction variables and **P_** variables with power (variables like $((T1*GPS)^2)$ are named with both, e.g. **P_2_I_T1_gps**). **LN_** represents logarithmic transformation. The user should check if in her dataset there are variables with these incipits. If it is the case, it is suggested to change the name of these variables before running the program.

*Dataset generated (gpsMDPolEst)*

One dataset named as specified in **file_pred()** including one row for each treatment point. The columns store the response, the partial derivatives, and the upper and lower bound of the confidence intervals.

If the dimensions are two, then, also nine datasets (where every result is presented in matrix form) are generated.
The results estimated from the sample has the names: `file_pred'Mat_response.dta, `file_pred'Mat_PD_` Dim1’.dta, `file_pred'Mat_PD_` Dim2’.dta

The names for the dataset including the bootstrap results in matrix form are similar, `file_pred'Mat_BootL_PD_` Dim1’, `file_pred'Mat_BootL_PD_` Dim2’ if the matrix stores the results for the lower bound, and `file_pred'Mat_BootH_PD_` Dim1’ `file_pred'Mat_BootH_PD_` Dim2’, if the matrix stores the results for the upper bound.

The matrixes include two columns with the names of the rows and columns for the easy implementation of plotmatrix.

**E-class objects generated (gpsMDPolEst)**

*macros:*

- `e(gpsmd)`: the name of the variable with the GPS estimates.
- `e(exogenous)`: the exogenous variables for the reduced equation estimation.
- `e(Dimensions)`: the dimensions of the treatment.
- `e(listgenvar)`: the program generates the variables as specified in model(string). This macro reports the list of the variables generated.
- `e(regmodel)`: the command for the regression for the polynomial estimation.
- `e(cmd)`: macro with the command.
- `e(cmdline)`: a macro with cmdline. This macro consists of the command just invoked and includes options and specifications.
- `e(Outcome)`: a macro containing the name of the outcome variable.

*matrices:*

- `e(matrtreat)`: a matrix with the points of the treatment for which the dose-response has been estimated.
- `e(returnresults)`: if matrixwithresults(T) the program return a matrix with the same information included in `file_pred’.dta.
7. A simple simulation for investigating the GPS method performance

In the rest of the article, we provide an example of the application of the package. Instead of proposing an application to a real dataset, we will show how the package works by using a generated dataset. This will enable me to investigate the performance of the GPS method by comparing the estimates obtained with GPS with those obtained using a rightly specified linear regression. The first step is, then, to generate the data. We set the following data generating process. The exogenous covariates are seven, $X_i \sim N(0,1); l = 1, ..., 7$, while the treatment dimensions are two, $T_1$ and $T_2$. The reduced equations are:

$$T_1 = 1 \cdot X_1 + 0.5 \cdot X_2 + 1 \cdot X_3 + 0.5 \cdot X_4 + 1 \cdot X_5 + 0.5 \cdot X_6 + 1 \cdot X_7 + \varepsilon_1$$
$$T_2 = 0.5 \cdot X_1 + 1 \cdot X_2 + 0.5 \cdot X_3 + 1 \cdot X_4 + 0.5 \cdot X_5 + 1 \cdot X_6 + 0.5 \cdot X_7 + \varepsilon_2$$

Where $\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix}$ follows a multivariate normal distribution:

$$\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix} \sim \mathcal{MN}(0_2, \Sigma)$$

$$\Sigma = \begin{bmatrix} 25 & 2 \\ 2 & 25 \end{bmatrix}$$

The variance-covariance matrix chosen, $\Sigma$, results in an $R$ squared of 0.80 and 0.75 when estimating the reduced equations for $T_1$ and $T_2$ respectively. The following code generates the data.

```
. clear all
. set more off
. set type double
. set matsize 11000
. *I set the seed.
. set seed 13131
. *I generate the observations.
. set obs 1200
. *I generate the exogenous covariates $X_m \ m=1, ..., 7$.
. gen X1 = 1 * rnormal(0,1)
. gen X2 = 2 * rnormal(0,1)
. gen X3 = 3 * rnormal(0,1)
. gen X4 = 4 * rnormal(0,1)
```
. gen X5 = 5 * rnormal(0,1)
. gen X6 = 6 * rnormal(0,1)
. gen X7 = 7 * rnormal(0,1)
. * I generate the residuals in the reduced equations and the treatments.
. * I define the matrix of correlation.
. matrix R = (25, 2 \ 2, 25)
. * I generate residuals from the multivariate normal.
. drawnorm V1 V2, cov(R)
. * I generate the treatments
. gen T1= 1*X1 + .5*X2 + 1*X3 + .5*X4 + 1*X5 + .5*X6 + 1*X7 + V1
. gen T2= .5*X1 + 1*X2 + .5*X3 + 1*X4 + .5*X5 + 1*X6 + .5*X7 + V2

The outcome follows the model:

\[ Y = 1 + 2 \cdot T_1 + 1.5 \cdot T_2 + 1 \cdot X_1 + 1.5 \cdot X_2 + 2 \cdot X_3 + 1 \cdot X_4 + 1.5 \cdot X_5 + 2 \cdot X_6 + 1 \cdot X_7 + \eta \]

Where \( \eta \sim \mathcal{N}(0,25) \). The R squared of this model, when estimated with a rightly specified linear regression, is 0.58.

. * I generate the outcome
. gen res= rnormal(0, 25)
. gen Y = 1+ 2*T1 + 1.5*T2 + 1*X1 + 1.5*X2 + 2*X3 + 1*X4 + 1.5*X5 + 2*X6 + 1*X7 + res

The first step involved in the implementation of the GPS method is to estimate the generalized propensity score \( G_i(T_{i},X_i) \). The command is gpsMD.

. * gpsMD
. gpsMD T1 T2, exogenous(X1 X2 X3 X4 X5 X6 X7) gpsMD(GPS)

**************
The regression for dimension: T1
**************

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>Number of obs</th>
<th>= 1,200</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F(7, 1192)</td>
<td>= 716.05</td>
</tr>
<tr>
<td>Model</td>
<td>123285.396</td>
<td>7</td>
<td>17612.1938</td>
<td>Prob &gt; F</td>
<td>= 0.0000</td>
</tr>
<tr>
<td>Residual</td>
<td>29318.8675</td>
<td>1,192</td>
<td>24.5963654</td>
<td>R-squared</td>
<td>= 0.8079</td>
</tr>
</tbody>
</table>
---

Adj R-squared = 0.8067

Total | 152604.224  1,199 127.27625  Root MSE = 4.9595

---

| T1 | Coef. | Std. Err. | t   | P>|t| | [95% Conf. Interval] |
|----|-------|-----------|-----|-----|----------------------|
|    | X1    | 1.260802  | 0.1459665 | 8.64 | 0.000 | 0.9744223 - 1.547182 |
|    | X2    | 0.5192245 | 0.0692367 | 7.50 | 0.000 | 0.3833851 - 0.6550639 |
|    | X3    | 0.9630556 | 0.0476939 | 20.19| 0.000 | 0.8694823 - 1.056629 |
|    | X4    | 0.4361229 | 0.0365548 | 11.93| 0.000 | 0.364404 - 0.5078417 |
|    | X5    | 1.002808  | 0.0203212 | 49.44| 0.000 | 0.9647476 - 1.044486 |
|    | X6    | 0.4845244 | 0.0240968 | 20.11| 0.000 | 0.4372475 - 0.5318013 |
|    | X7    | 1.004617  | 0.0203212 | 49.44| 0.000 | 0.9647476 - 1.044486 |
|    | _cons | -0.1534133| 0.1439645 | -1.07| 0.287 | -0.4358655 - 0.1290388 |
---

The regression for dimension: T2

Source | SS       | df | MS          | Number of obs = 1,200
--------|----------|----|-------------|-------------------|
Model   | 96776.5041 | 7  | 13825.2149  | F(7, 1192) = 524.03 |
Residual| 31448.1284 | 1192| 26.3826581  | Prob > F = 0.0000 |
Total   | 128224.632 | 1199| 106.94298   | R-squared = 0.7547 |
Adj R-squared = 0.7533

---

| T2 | Coef. | Std. Err. | t   | P>|t| | [95% Conf. Interval] |
|----|-------|-----------|-----|-----|----------------------|
|    | X1    | 0.6615431 | 0.1511739 | 4.38 | 0.000 | 0.3649465 - 0.9581397 |
|    | X2    | 0.8111875 | 0.071068 | 11.31| 0.000 | 0.6705019 - 0.9518731 |
|    | X3    | 0.509123  | 0.0493954 | 10.31| 0.000 | 0.4122114 - 0.6060346 |
|    | X4    | 0.9728744 | 0.0378589 | 25.70| 0.000 | 0.8985969 - 1.047152 |
|    | X5    | 0.5054108 | 0.0296139 | 17.07| 0.000 | 0.4473096 - 0.563512 |
|    | X6    | 1.036732  | 0.0249565 | 41.54| 0.000 | 0.9877685 - 1.085696 |
|    | X7    | 0.4989273 | 0.0210461 | 23.71| 0.000 | 0.4576357 - 0.5402189 |
|    | _cons | 0.0161213 | 0.1491006 | 0.11 | 0.914 | -0.2764075 - 0.3086501 |
---
The Variance Covariance Matrix:

<table>
<thead>
<tr>
<th></th>
<th>T1Res</th>
<th>T2Res</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1Res</td>
<td>24.45277</td>
<td></td>
</tr>
<tr>
<td>T2Res</td>
<td>1.308303</td>
<td>26.22863</td>
</tr>
</tbody>
</table>

The command generates a variable named “GPS” and a printed output consisting of the outputs for the regressions estimating the reduced equations, and the variance-covariance matrix.

The second step consists in identifying the observations outside the common support. The command is ComSupp. The output consists of a new dummy variable taking value 1 if the observation is outside the common support and 0 otherwise. The one-way table of frequencies of this variable is printed.

The choice of the cutpoints is somehow arbitrary and affects the number of observations in the common support. We chose to discretize each dimension in two intervals corresponding to four discrete treatments.

. * ComSupp
. ComSupp T1 T2, exogenous(X1 X2 X3 X4 X5 X6 X7) index(“p50”) cutpoints(2) obs_notsup(Commonsupport)

COMMON SUPPORT (variable: "Commonsupport")
1 correspond to observations outside the common support
0 correspond to observations inside the common support

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>739</td>
<td>61.58</td>
<td>61.58</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>461</td>
<td>38.42</td>
<td>100.00</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>1,200</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>
The reported results show that only 61.58\% of the sample is within the common support region. Unreported simulations resulted in restrictions with a similar magnitude (see also the restriction applied in Egger et al. (2020)). It seems that finding a wide common support is more demanding with multidimensional treatments than with mono-dimensional treatments. Although restricting the sample to the common support increases the consistency and credibility of the estimates, it can also be problematic when important observations are excluded (Lechner 2008). In the conclusion, we will propose a strategy that, arguably, can be adopted to reduce this flaw.

The third step consists in evaluating the balancing property. We restrict the evaluation to the common support region specifying the option \texttt{obs\_notsup}. As in the case of \texttt{ComSupp}, we specify dividing the treatments into four sets. The propensity score is instead divided into four intervals (\texttt{nq\_gpsMD(4)}).

```
*. * gpsMDbal
  . gpsMDbal X1 X2 X3 X4 X5 X6 X7, index("p50") cutpoints(2) nq_gpsMD(4) discrTreat(Discretetreat) obs_not sup(Commonsupport)

***************
In the following TabellaImpRes is reported
***************

<table>
<thead>
<tr>
<th>1r(p)      2r(p)      3r(p)      4r(p)  1Adj_r(p)  2Adj_r(p)  3Adj_r(p)  4Adj_r(p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
</tr>
<tr>
<td>X2</td>
</tr>
<tr>
<td>X3</td>
</tr>
<tr>
<td>X4</td>
</tr>
<tr>
<td>X5</td>
</tr>
<tr>
<td>X6</td>
</tr>
<tr>
<td>X7</td>
</tr>
</tbody>
</table>

***************
In the following TabellaImpRes is reported but p-values higher than the threshold specified in level(string) are omitted
***************

<table>
<thead>
<tr>
<th>1r(p)      2r(p)      3r(p)      4r(p)  1Adj_r(p)  2Adj_r(p)  3Adj_r(p)  4Adj_r(p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
</tr>
</tbody>
</table>
For simplifying reading, the table with the output is printed with and without omitting cells whose p-value is higher than the threshold decided by the user (in this case the default: 0.05). In this case, we see that before adjusting for GPS only $X_1, X_2, X_3$ were reasonably balanced. Adjusting for GPS removes the unbalance in 13 groups over 17 and thus increases quite strongly the balance in the dataset.

The last step involves the estimation of the control function and the response corresponding to the chosen doses. Also here, we restrict the estimation to the common support region. We chose not to specify any particular matrix of the doses for which we intend to estimate the corresponding response. We instead divide the range of every dimension into 3 intervals. The Cartesian product of the extremes of the intervals corresponds to the 16 doses for which the program estimates the response. It is worth noticing that when only the common support region is considered (as in this example), the program chooses the doses considering only the observations on the common support. As suggested in these cases (Efron and Tibshirani 1994), to obtain the confidence intervals for the estimated response, we bootstrap each estimation 1000 times. We specify both index and cutpoints for constraining the estimation to the common support region.

The command prints the result of the regression estimation. However, it is worth noticing that coefficients have not any causal interpretation (Hirano and Imbens 2004; Bia and Mattei 2008) and that the various tests reported are biased because GPS is a generated regressor (Wooldridge 2010). Only testing if terms including GPS are jointly different from 0 would be informative since it could be considered as a test of whether exogenous covariates introduce bias (Hirano and Imbens 2004; Bia and Mattei 2008).

```
. * gpsMDPolEst
. gpsMDPolEst Y T1 T2, gpsmd(GPS) exogenous(X1 X2 X3 X4 X5 X6 X7) ///
> model("T1 + T2 + GPS + T1*GPS + T2*GPS") ///
> file_pred(ExampleStata) numboot(1000) dividingint(3) index("p50") cutpoints(2)

**************
The regression estimating the dose-response function is calculated only on the common support. The output is the following:
**************

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>Number of obs</th>
<th>739</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F(5, 733)</td>
<td>182.20</td>
</tr>
</tbody>
</table>
```
We now compare the result of the GPS method with those obtained by using a rightly specified linear regression. We constrain the sample to the common support region. Figure 1 reports the responses and the upper and lower bounds for the estimations obtained with both linear regression and the GPS method. Confidence intervals are calculated at the 0.95 level.

The estimated responses obtained by using the GPS method are qualitatively and quantitatively similar to those obtained by using a rightly specified regression. Indeed, they are statistically equal. This suggests that the GPS method succeeds in reducing bias.

```
. *comparison with a regression
. mat define matrtreat= e(matrtreat)
. mat define returnresults=e(returnresults)
. qui: reg c.Y c.T1 c.T2 c.X1 c.X2 c.X3 c.X4 c.X5 c.X6 c.X7 if Commonsupport==0
. qui: margins, at( T1= `=matrtreat[1,1]' T2= `=matrtreat[1,2]' )
```
. mat define temp=r(table)
* mat define resreg=('matrtreat[1,1]', 'matrtreat[1,2]', temp[5,1], temp[1,1], temp[6,1])
. forvalues i=2(1)`=rowsof(matrtreat)' {
    . qui: margins, at(T1='matrtreat[`i',1]' T2='matrtreat[`i',2]' }
. mat define temp=r(table)
. mat define resreg=(resreg 
    ('matrtreat[`i',1]', 'matrtreat[`i',2]', temp[5,1], temp[1,1], temp[6,1]))
. mat colnames resreg = T1 T2 LBReg ResponseReg UBReg
. mat define resreg=(resreg, returnresults[1...,9], returnresults[1...,3] ,returnresults[1...,6])

*I generate the graph
. clear
. set obs 16
. svmat resreg, names(col)
. gen treatment_levels="(" + string(T1, "%9.2f") + "," + string(T2, "%9.2f") + ")"
. encode treatment_levels, gen("id")
. label variable id "Treatment levels"
. label variable ResponseReg "Reg. Response"
. label variable LBReg "Reg. Low. Bound"
. label variable UBReg "Reg. Up. Bound"
. label variable Response "GPS. Response"
. label variable BootL_response "GPS. Low. Bound"
. label variable BootH_response "GPS. Up. Bound"

. graph twoway (scatter ResponseReg id, mcolor(blue) msymbol(circle) msize(small)) ///
    > (scatter UBReg id , mcolor(blue) msymbol(plus) msize(small)) ///
    > (scatter LBReg id , mcolor(blue) msymbol(circle_hollow) msize(small)) ///
    > (scatter Response id , mcolor(red) msymbol(triangle) msize(small)) ///
    > (scatter BootL_response id , mcolor(red) msymbol(triangle_hollow ) msize(small)), ///
    > title("Regression and GPS estimations correspondence") xtitle("Treatment levels (T1,T2)") ///
    > ytitle("E[Y(t)]", size(med)) ///
    > xlabel(1(1)16 , labsize(small) valuelabel angle(45)) ///
    > ylabel(), labsize(small) valuelabel angle(0)) ///
    > legend(colfirst) ///
    > name(CompGPSReg, replace)
Figure 1 Regression and GPS estimations. For both Regression estimates and GPS estimates: Point estimate, Upper bound, Lower bound. Bounds are calculated at the 0.95 confidence level.

8. Conclusions and a proposal for overcoming small common support

Causal evaluation has been generally restricted to binary or continuous treatments. In this paper, we present a novel Stata package, gpsMD, which implements Egger and von Ehrlich’s (2013) extension of the GPS method to the cases when treatment is multidimensional and continuous. After having summarized the econometric framework and described the commands, we present a simple simulated dataset in order for the reader to familiarize herself with the commands. Moreover, we compare the results obtained with the GPS method with those obtained employing a linear regression rightly specified. As expected, the dose-response estimated with the GPS method is not statistically different from those obtained with the regression, thus suggesting that the GPS method is effective in estimating causal parameters in the case of multidimensional continuous treatments. Nevertheless, the implementation evidences also a potential problem in the application of the GPS method: the lack of common support. Although restricting the sample improves the reliability of the estimation it may exclude from the analysis interesting observations.

The criterion used to define the common support exploits the entire range of the treatment that is observed in the sample. At first, we divide the treatment into an arbitrary number of subsets, then by assuming that
every arbitrary subset can be meaningfully represented by a point (mean or median) we transform the continuous treatment into a multi-valued treatment. Second, for each value of the treatment, we calculate the density of obtaining the value given the covariates and we remove observations with a pattern of covariates whose corresponding density is not – in the sample – sufficiently represented in both the treated (the set of observation that has a treatment falling in the interval represented by the value) and the control (all the other observations) group. The observations that remain constitute the common support region and our estimate of the dose-response function is reliable for this group only. Generally, the analysis is also constrained to the treatment range in the resulting sample.

In these conclusions, we propose that in order to include the observations which are not in the common support in our analysis we can act iteratively. Once we have done the first selection, we can then use the range of the treatment of the highest [lowest] part of the distribution (which is a subset of the starting range) and finding a common support considering only that range. The further analysis can only generalize to the resulting common support regions and the range of treatment that results. It is worth noting that although the selection is led by a restriction on the treatment the final selection of the sample is based on exogenous covariates. The rationale of the criterion proposed is that the observations outside the common support region calculated in the first step present patterns of observables that theoretically almost impede (while empirically simply impede) to have a treatment belonging to a given interval and, therefore, empirically is not unreasonable to treat them as a sample from a population on his own. The continuous treatment would, therefore, be divided into a family of intervals empirically considered qualitatively different from each other, and for any of those only a subset of the original population can be eligible.
9. References


Guo, Shenyang, and Mark W. Fraser. 2015. Propensity Score Analysis. SAGE.


Mander, Adrian. 2019. ‘PLOTMATRIX: Stata Module to Plot Values of a Matrix as Different Coloured Blocks’. Boston College Department of Economics.


Rostam-Afschar, Davud, and Robin Jessen. 2014. ‘GRAPH3D: Stata Module to Draw Colored, Scalable, Rotatable 3D Plots’. Boston College Department of Economics.

11. Appendix 1

Proof of the Balancing property, Theorem 1 and 2 (Peter H. Egger and von Ehrlich 2013)

Before the various proofs, we report Theorem 2.1.8 (Casella and Berger 2002, 2:53) for the transformation of random variables. Proofs assume the propensity score is a function satisfying the premises of Theorem 2.1.8.

Theorem 2.1.8 (Casella and Berger 2002, 2:53): Let $X$ have a pdf $f_X(x)$, let $Y = g(X)$, and define the sample space $\mathcal{X}$ as $\mathcal{X} = \{x: f_X(x) > 0\}$. Suppose there exists a partition, $A_0, A_1, \ldots, A_k$, of $\mathcal{X}$ such that $P(X \in A_0) = 0$ and $f_X(x)$ is continuous on each $A_i$. Further suppose there exists functions $l_1(x), \ldots, l_k(x)$ defined on $A_1, \ldots, A_k$, respectively satisfying:

i. $l(x) = l_i(x)$, for $x \in A_i$
ii. $l_i(x)$ is monotone on $A_i$
iii. the set $\mathcal{Y} = \{y: y = l_i(x) \text{ for some } x \in A_i\}$ is the same for each $i = 1, \ldots, k$

and

iv. $l_i^{-1}(y)$ has a continuous derivative on $\mathcal{Y}$, for each $i = 1, \ldots, k$

then

$$f_Y(y) = \begin{cases} \sum_{i=1}^{k} f_X(l_i^{-1}(y)) \left| \frac{dl_i^{-1}(y)}{dy} \right| & y \in \mathcal{Y} \\ 0 & \text{otherwise} \end{cases}$$

To simplify the notation we suppress the subscript for the individuals in this section. We denote probability density functions as $f(.)$. We suppress the subscripts indicating the random variable from the notation because it is obvious from the context (e.g. $f_Y(y) = f(y)$).

Lemma 1: $f(t|f(t|Z) = k) = k$

Proof
Balancing property: $f(t|Z, g(t, Z)) = f(t| g(t, Z))$

Proof

$$f(t|Z, g(t, Z)) = f(t|Z) = f(t| f(t|Z)) = f(t| g(t, Z))$$

Theorem 1: $Y_i(t) \perp T_i| g(t, Z), \forall t \in \mathbb{Z}$

Proof

We need to show that $f(t| g(t, Z), Y(t)) = f(t| g(t, Z))$. Then we show that both sides are equal to $g(t, Z) = k$

a)
Similarly to Theorem 2 in Rosenbaum and Rubin

\[ f(t | g(t, Z) = k, Y(t)) = \frac{f(t, f(t | Z), Y(t))}{f(f(t | Z), Y(t))} = \frac{\sum_{i:f(t | Z) = k} f\left(t, l_i^{-1}(f(t | Z)), Y(t)\right) \left| \frac{dl_i^{-1}(f(t | Z))}{df(t | Z)} \right|}{\sum_{i:f(t | Z) = k} f\left(l_i^{-1}(f(t | Z)), Y(t)\right) \left| \frac{dl_i^{-1}(f(t | Z))}{df(t | Z)} \right|} \]

\[ = \frac{\sum_{i:f(t | Z) = k} f\left(t | l_i^{-1}(f(t | Z)), Y(t)\right) f\left(l_i^{-1}(f(t | Z)), Y(t)\right) \left| \frac{dl_i^{-1}(f(t | Z))}{df(t | Z)} \right|}{\sum_{i:f(t | Z) = k} f\left(l_i^{-1}(f(t | Z)), Y(t)\right) \left| \frac{dl_i^{-1}(f(t | Z))}{df(t | Z)} \right|} \]

\[ = \frac{\sum_{i:f(t | Z) = k} f\left(l_i^{-1}(f(t | Z)), Y(t)\right) \left| \frac{dl_i^{-1}(f(t | Z))}{df(t | Z)} \right|}{\sum_{i:f(t | Z) = k} f\left(l_i^{-1}(f(t | Z)), Y(t)\right) \left| \frac{dl_i^{-1}(f(t | Z))}{df(t | Z)} \right|} = k \]

\[ = k \]

**Proof**

See Egger and von Ehrlich (2013)

**Theorem 2 (Peter H. Egger and von Ehrlich 2013):** Denote \( \mu(t, g) \equiv E[Y(t) | g(t, Z)] \). Under weak unconfoundedness:

i. \( \mu(t, g) \equiv E[Y(t) | g(t, Z) = k] = E[Y | T = t, G = k] \)

ii. \( \mu(t) = E_{g(t,Z)}[\mu(t, g(t, Z))] \)

**Proof that a balancing function works well as the propensity score**

Let \( b(T, Z) \) a function such that the balancing property is satisfied. This means:

\[ f(t | Z, b(t, Z)) = f(t | b(t, Z)) \]

**Lemma 1 – Balancing function:** \( b(t, z) = b(t, m) \Rightarrow g(t, z) = g(t, m) \)

**Proof**

Similarly to Theorem 2 in Rosenbaum and Rubin (1983).
By absurd, if \( g(t, z) \neq g(t, m) \) and \( b(t, z) = b(t, m) \), then \( f(t|z) \neq f(t|m) \) and \( b(t, Z) \) cannot balance.

Indeed if \( b(t, m) = b(t, z) \) and \( b(t, Z) \) is a balancing function it cannot be that:

\[
\begin{align*}
  f(t|b(t, z)) &= f(t|z, b(t, z)) = f(t|z) = g(t, z) \neq g(t, m) = f(t|m) = f(t|m, b(t, m)) \\
  &= f(t|b(t, m))
\end{align*}
\]

With a balancing function the theorems become:

**Theorem 1 - balancing function:** \( Y_i(t) \perp T_i|b(t, Z_i), \forall t \in \mathcal{Z} \)

**Proof**

We need to show that \( f(t|b(t, Z), Y(t)) = f(t|b(t, Z)) \). Then we show that both sides are equal to \( g(t, Z) = g \).

a)

\[
\begin{align*}
  f(t|b(t, Z) = k, Y(t)) &= \frac{f(t, b(t, Z), Y(t))}{f(b(t, Z), Y(t))} = \sum_{i:b(t,Z)=k} f \left( t, l_i^{-1}(b(t, Z)), Y(t) \right) |\frac{dl_i^{-1}(b(t, Z))}{db(t, Z)}| \\
  &= \sum_{i:b(t,Z)=k} f \left( t | l_i^{-1}(b(t, Z)), Y(t) \right) f \left( l_i^{-1}(b(t, Z)), Y(t) \right) |\frac{dl_i^{-1}(b(t, Z))}{df(t|Z)}| \\
  &= g \sum_{i:b(t,Z)=k} f \left( l_i^{-1}(b(t|Z)), Y(t) \right) |\frac{dl_i^{-1}(b(t|Z))}{df(t|Z)}| \\
  &= g
\end{align*}
\]

b)

\[
\begin{align*}
  f(t|b(t, Z) = k) &= f(t|Z, b(t, Z)) = f(t|Z) = g
\end{align*}
\]

Theorem 2 is similar as in the case of the propensity score.
12. Appendix 2

_Akaikemax_

The command identifies the combination of variables that minimizes the chosen information criterion. Only linear regression models are supported. It works simply by brutal force. Given the rules specified in the options, the list of all the possible combinations of variables is generated. Then, by using each combination of variables a regression is run and the information criterion computed. Finally, the model that minimizes the criterion is chosen.

_The Syntax_

Akaikemax, outcome(varlist max=1) power(numlist integer max=1) [baseVar(varlist min=1) controls(varlist) noIntNoPow(varlist) aloneandpow(varlist) aloneVars(varlist) ic(string) stopiflarge(string) reg_opt(string)]

outcome (varlist max=1): the outcome variable.

baseVar (varlist min=1): the variables that it is required to be always in the model. These variables are also interacted with controls and/or exponentiated as indicated in power.

power (numlist integer max=1): the power to which interactions (if needed) and basevars will be exponentiated.

controls (varlist): the variables that will be interacted and exponentiated with BaseVars (if you want also the variables alone you should specify them in noIntNoPow or aloneandpow or aloneVars). If specified, also baseVar must be specified.

aloneandpow(varlist): Variables that will enter alone and exponentiated in the model.

noIntNoPow (varlist): Variables that will enter in the combinations simply as they are (no interacted nor exponentiated).

aloneVars (varlist): Variables that will always enter in the model simply as they are (neither interacted nor exponentiated).

ic(string): the information criteria AIC (default) or BIC.

stopiflarge(string): the combinations can be unexpectedly many. The default is that the program stops if the number of combinations is higher than 2^20. If set to "N" the program does not stop and only the pc of the user or Stata limits themselves have value.

reg_opt(string): a string with all the options the user would put in regress (e.g. no constant).
Variables generated

The command generates variables for the interaction and the exponentiation (Only those variables belonging to the preferred model will be kept). They all start with I_ or P_.

e-class object

Macro

e(bestmodel) : the command for the linear regression with the minimum information criterion.
e(ic) : the information criterion chosen.
e(cmd) : the name of the command just launched.
e(cmdline) : The command just launched.
e(NAattempt) : The number of combinations tested.
e(NewVarInBest) : a list with the variables in the best model which are generated by the program.

Scalar

e(minIC) : scalar with the Information criterion value of the best model.
13. Appendix C

Derivation of the t-test for the balancing property

The sample quantity is:

\[ \frac{1}{N} \sum_{g(\bar{d},z)} N g(\bar{d},z)^D \left( \bar{z}_D g(\bar{d},z)^D - \bar{z}_D g(\bar{d},z)^D \right) \]

With a change of notation, it becomes:

\[ \frac{1}{N} \sum_{j} N_j (\bar{X}_j - \bar{Y}_j) \]

Where \( \bar{z}_D g(\bar{d},z)^D = \bar{X}_j, \bar{z}_D g(\bar{d},z)^D = \bar{Y}_j \) for \( j = 1, ..., K \) the index for the intervals of the generalized propensity score. Define \( N_{X_j}, N_{Y_j} \) respectively the number of units in the sets \( X_j \) and \( Y_j \). By the central limit theorem, see Ross (2004), for large enough \( N_{X_j}, N_{Y_j} \) the following holds:

\[ \bar{X}_j \sim N \left( \mu_{X_j}, \frac{\sigma_{X_j}^2}{N_{X_j}} \right), j = 1, ..., K \]

\[ \bar{Y}_j \sim N \left( \mu_{Y_j}, \frac{\sigma_{Y_j}^2}{N_{Y_j}} \right), j = 1, ..., K \]

Then, since sets are assumed independent:

\[ \frac{N_j}{N} \bar{X}_j \sim N \left( \frac{N_j}{N} \mu_{X_j}, \left( \frac{N_j}{N} \right)^2 \frac{\sigma_{X_j}^2}{N_{X_j}} \right), j = 1, ..., K \]

\[ \frac{N_j}{N} \bar{Y}_j \sim N \left( \frac{N_j}{N} \mu_{Y_j}, \left( \frac{N_j}{N} \right)^2 \frac{\sigma_{Y_j}^2}{N_{Y_j}} \right), j = 1, ..., K \]

It follows that the distribution of the sum of the \( K \) couples is then:

\[ \sum_{j} \frac{N_j}{N} (\bar{X}_j - \bar{Y}_j) \sim N \left( \sum_{j} \frac{N_j}{N} (\mu_{X_j} - \mu_{Y_j}), \sum_{j} \left[ \left( \frac{N_j}{N} \right)^2 \left( \frac{\sigma_{X_j}^2}{N_{X_j}} + \frac{\sigma_{Y_j}^2}{N_{Y_j}} \right) \right] \right) \]

And thus:
\[ \frac{\sum_j^K \frac{N_j}{N} (\bar{X}_j - \bar{Y}_j) - \sum_j^K \frac{N_j}{N} (\mu_{X_j} - \mu_{Y_j})}{\sqrt{\sum_j^K \left[ \frac{N_j}{N} \left( \frac{\sigma^2_{X_j}}{N_{X_j}} + \frac{\sigma^2_{Y_j}}{N_{Y_j}} \right) \right]}} \sim N(0, 1) \]

The estimators for the variances \( \sigma^2 \) are

\[ S^2_{X_j} = \frac{1}{N_{X_j} - 1} \sum_{i=1}^{N_{X_j}} (X_{ji} - \bar{X}_j)^2, j = 1, \ldots, K \]

\[ S^2_{Y_j} = \frac{1}{N_{Y_j} - 1} \sum_{i=1}^{N_{Y_j}} (Y_{ji} - \bar{Y}_j)^2, j = 1, \ldots, K \]

The well-known result (Ross 2004) is

\[ \frac{(N_{X_j} - 1)}{\sigma^2_{X_j}} S^2_{X_j} \sim \chi^2_{N_{X_j} - 1}, j = 1, \ldots, K \]

\[ \frac{(N_{Y_j} - 1)}{\sigma^2_{Y_j}} S^2_{Y_j} \sim \chi^2_{N_{Y_j} - 1}, j = 1, \ldots, K \]

Then

\[ \sum_j^K \left( \frac{(N_{X_j} - 1)}{\sigma^2_{X_j}} S^2_{X_j} + \frac{(N_{Y_j} - 1)}{\sigma^2_{Y_j}} S^2_{Y_j} \right) \sim \chi^2_{N - 2K} \]

If we assume that \( \sigma^2_{X_j} = \sigma^2_{X_j} = \sigma^2_{Y_j} = \sigma^2_{Y_j} = \sigma^2 \):

\[ \frac{\sum_j^K \frac{N_j}{N} (\bar{X}_j - \bar{Y}_j) - \sum_j^K \frac{N_j}{N} (\mu_{X_j} - \mu_{Y_j})}{\sqrt{\sum_j^K \left[ \frac{N_j}{N} \left( \frac{\sigma^2_{X_j}}{N_{X_j}} + \frac{\sigma^2_{Y_j}}{N_{Y_j}} \right) \right]}} = \frac{1}{N} \sum_j^K N_j (\bar{X}_j - \bar{Y}_j) - \sum_j^K N_j (\mu_{X_j} - \mu_{Y_j}) \]

\[ \sqrt{\sum_j^K \left[ \frac{N_j}{N} \left( \frac{\sigma^2_{X_j}}{N_{X_j}} + \frac{\sigma^2_{Y_j}}{N_{Y_j}} \right) \right] \sqrt{\sum_j^K \left[ \frac{N_j}{N} \left( \frac{1}{N_{X_j}} + \frac{1}{N_{Y_j}} \right) \right]}} \]

\[ \sum_j^K \left( \frac{(N_{X_j} - 1)}{\sigma^2_{X_j}} S^2_{X_j} + \frac{(N_{Y_j} - 1)}{\sigma^2_{Y_j}} S^2_{Y_j} \right) = \frac{1}{\sigma^2} \sum_j^K \left[ (N_{X_j} - 1) S^2_{X_j} + (N_{Y_j} - 1) S^2_{Y_j} \right] \]

Then:
\[
\frac{\sum_j^K N_j (\bar{X}_j - \bar{Y}_j) - \sum_j^K N_j \left( \mu_{X_j} - \mu_{Y_j} \right)}{\sigma \sqrt{\sum_j^K N_j^2 \left( \frac{1}{N_{X_j}} + \frac{1}{N_{Y_j}} \right)}} \cdot \frac{\sqrt{\frac{\sigma^2 (N - 2K)}{\sum_j^K \left( N_{X_j} - 1 \right) S_{X_j}^2 + \left( N_{Y_j} - 1 \right) S_{Y_j}^2}}}{\sqrt{\sum_j^K \left( N_{X_j} - 1 \right) S_{X_j}^2 + \left( N_{Y_j} - 1 \right) S_{Y_j}^2}} = \]
\[
\frac{\sum_j^K N_j (\bar{X}_j - \bar{Y}_j) - \sum_j^K N_j \left( \mu_{X_j} - \mu_{Y_j} \right)}{\sqrt{\sum_j^K N_j^2 \left( \frac{1}{N_{X_j}} + \frac{1}{N_{Y_j}} \right)}} \cdot \frac{(N - 2K)}{\sum_j^K \left( N_{X_j} - 1 \right) S_{X_j}^2 + \left( N_{Y_j} - 1 \right) S_{Y_j}^2} \sim t_{N - 2K}
\]

The p-value is then\(^6\)

\[
pvalue = 2 \cdot \left\{ 1 - F_{t_{N - 2K}} \left( \frac{\sum_j^K N_j (\bar{X}_j - \bar{Y}_j) - 0}{\sqrt{\sum_j^K N_j^2 \left( \frac{1}{N_{X_j}} + \frac{1}{N_{Y_j}} \right)}} \cdot \frac{(N - 2K)}{\sum_j^K \left( N_{X_j} - 1 \right) S_{X_j}^2 + \left( N_{Y_j} - 1 \right) S_{Y_j}^2} \right) \right\}
\]

\(^6\) In order to avoid precision problems, the actual formula exploits the fact that the t distribution is symmetric (Gould 2006)