Finanziato dall'Unione europea
NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

Consiglio Nazionale delle Ricerche

Data-driven decision making using Stata

UK Stata Conference 2024
London, LSE
12-13 September 2024

FOSSR
Fostering Open Science in Social Science Research
Innovative tools and services to investigate economic and societal change

Giovanni Cerulli
CNR – IRCRES
National Research Council of Italy
Research Institute on Sustainable Economic Growth


Next
Generation
EU
#NextGenerationEU
#EUBudget
European Commission


RECOVERY AND RESILIENCE FACILITY
THE 6 PRIORITIES
1 GREEN TRANSITION
2 DIGITAL TRANSFORMATION
3 ECONOMIC COHESION, PRODUCTIVITY AND COMPETITIVENESS
4 SOCIAL AND TERRITORIAL COHESION
5 INSTITUTIONAL RESILIENCE
6 POLICIES FOR THE NEXT GENERATION
#EUSolidarity #StrongerTogether

32 million project

FOSSR
Fostering Open Science in Social Science Research
Innovative tools and services to investigate economic and societal change

Consiglio Nazionale delle Ricerche IRCrES ISTITUTO di RICERCA sulla CRESCITA ECONOMICA SOSTENIBILE RESEARCH INSTITUTE on SUSTAINABLE ECONOMIC GROWTH
IRPPS
CNR ISTC
ISTI
ICAR CNR
CNR ISMed
DSU Dipartimento Scienze Umane e Sociali Patrimonio Culturale
Consiglio Nazionale delle Ricerche Centro Interdipartimentale per l'Etica e l'Integrità nella Ricerca
Consiglio Nazionale delle Ricerche Unità Valorizzazione della Ricerca

Machine Learning
(prediction)

Causal Inference
(counterfactual)

Data-driven Decision Making
(Optimal Policy Learning, OPL)

DEFINITION OF OPL

- **What is policy learning?**

  Process of improving program **welfare** achievements by re-iterating similar policies over time

- **Optimal treatment assignment**

  Policymakers can **optimally fine-tune the treatment assignment** of a prospective policy using the results from an RCT or observational study. Assignment rules depends on the **class of policies** considered (here we focus on **threshold-based** and **linear-combination** policies)

- **Maximizing constrained welfare**

  The policymaker hardly manage to reach the best solution (**unconstrained maximum welfare**) because of institutional/economic contains of various sort

Background literature

Athey, S., and S. Wager. 2021. "Policy Learning with Observational Data." Econometrica 89 (1): 133–161.

Bhattacharya, D., and P. Dupas. 2012. "Inferring Welfare Maximizing Treatment Assignment under Budget Constraints." Journal of Econometrics 167 (1): 168–196.

Dehejia, R. 2005. "Program Evaluation as a Decision Problem." Journal of Econometrics 125 (1–2): 141–173.

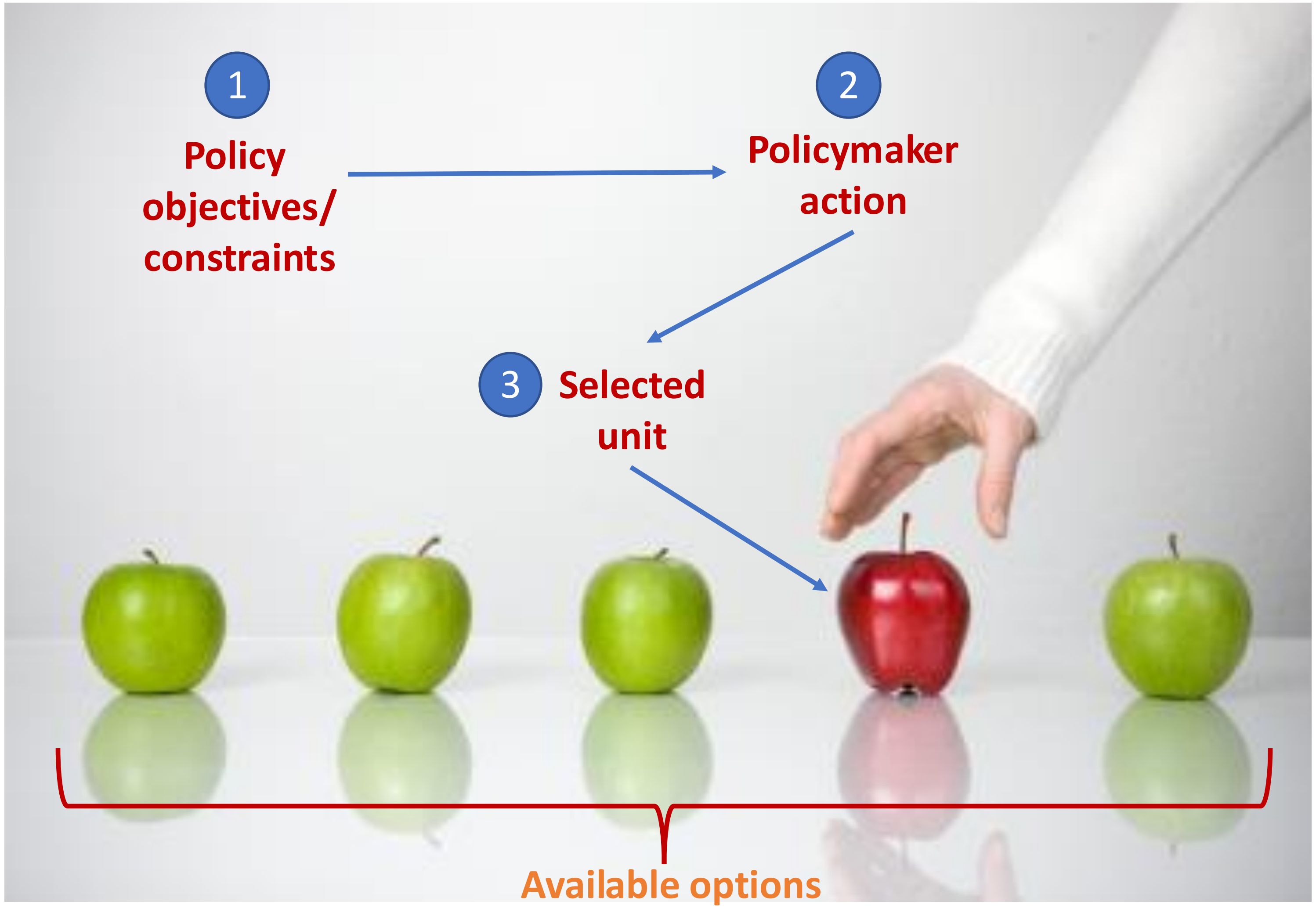
Hirano, K., and J. R. Porter. 2009. "Asymptotics for Statistical Treatment Rules." Econometrica 77 (5): 1683–1701.

Kitagawa, T., and A. Tetenov. 2018. "Who Should Be Treated? Empirical Welfare Maximization Methods for Treatment Choice." Econometrica 86 (2): 591–616.

Manski, C. F. 2004. "Statistical Treatment Rules for Heterogeneous Populations." Econometrica 72 (4):

Zhou, Z., S. Athey, and S. Wager. 2018. "Offline Multi-Action Policy Learning: Generalization and Optimization." arXiv Preprint arXiv. 1810.04778.

Policy as a *selection problem*



Policy direct and indirect effect

Total Policy effect

$$\gamma(S) = \underbrace{\alpha}_{\text{Direct effect}} + \underbrace{\beta(S)}_{\text{Indirect effect}}$$

S = Selection process operated by a specific **treatment rule**

Empirical Welfare Maximization aims at maximizing the **indirect effect** via *optimal assignment to treatment* within specific *classes of policies*

This is the effect obtained if the "assignment to treatment" was run at **random**

Optimal treatment assignment

Let X be an individual's vector of characteristics, Y an outcome of interest, $T = \{0, 1\}$ a binary treatment. A policy assignment rule $\mathcal{G}$ is a function mapping X to T, specifying which individuals are or are not to be treated:

$$\mathcal{G} : X \to T$$

Define the (population) policy conditional average treatment effect as:

$$\tau(X) = E(Y_1|X) - E(Y_0|X)$$

where Y_1 and Y_0 represent the two potential outcomes of the policy, and $E_X[\tau(X)] = \tau$ the average treatment effect.

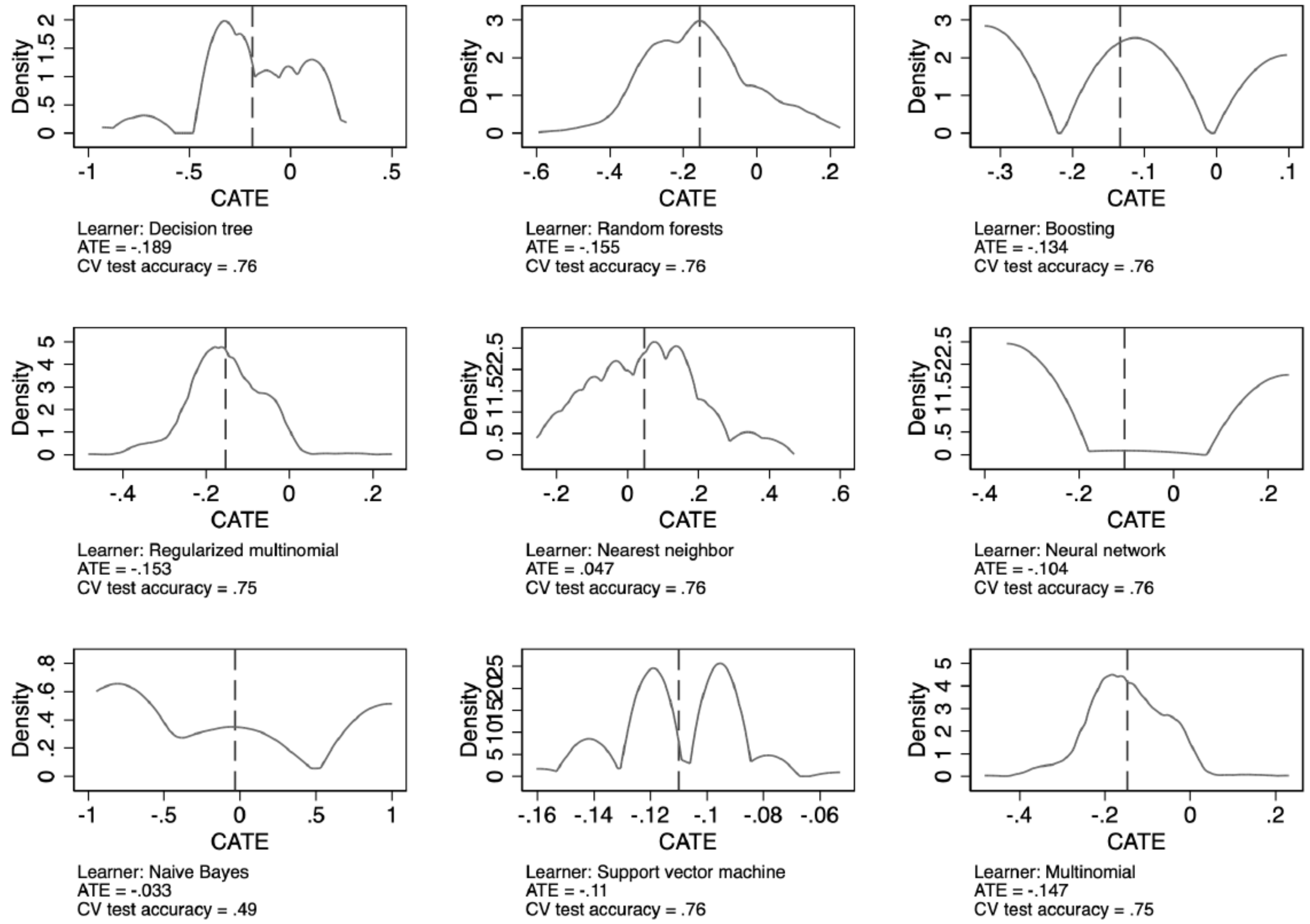
Under **selection-on-observables**, we know that:

$$\tau(X) = E(Y|X,T=1) - E(Y|X,T=0)$$

These two **conditional expectations** are **identified** by data. Whatever **ML algorithm** can be used for estimation (Boosting, Random forests, Neural networks, Nearest neighbor, etc.)

Extension to **selection-on-unobservables** straightforward

ML estimation of $\tau(X)$



Estimation of the **distribution** of the **conditional average treatment effects** (**CATE**) using the ML methods implemented via **`c_ml_stata_cv`** (Cerulli, 2022). Note: dashed vertical line indicates the **average treatment effect** (**ATE**).

Finanziato dall'Unione europea
NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

Consiglio Nazionale delle Ricerche

Optimal treatment assignment and *regret* estimation

The estimated policy actual total effect (or *welfare*)

$$\widehat{W} = \sum_{i=1}^{N} T_i \cdot \hat{\tau}(X_i)$$

and the estimated policy *unconstrained* optimal total effect (or *unconstrained maximum welfare*) as:

$$\widehat{W}^* = \sum_{i=1}^{N} \hat{T}_i^* \cdot \hat{\tau}(X_i)$$

where:

$$\hat{T}_i^* = \mathbf{1}[\hat{\tau}(X_i) > 0]$$

is the estimated optimal unconstrained policy assignment.

The difference between the estimated (unconstrained) maximum achievable welfare and the estimated welfare associated to the policy actually run is called *regret*, and it is defined as:

$$\widehat{regret} = \widehat{W}^* - \widehat{W}$$

NAÏVE OPTIMAL SELECTION

1. Given {*X*,*Y*,*T*} from an already-implemented policy: estimate the **idiosyncratic effect $\boldsymbol{\tau(X)}$**. This means we have learnt the mapping:

$$X \to \tau(X) \quad \textit{(learning from experience)}$$

2. Consider a prospective second policy round with a new eligible set {*X’*}, and compute the learnt $\{\tau(X')\}$ over *X’*.

3. Rank individuals so that: $\tau(X_1') > \tau(X_2') > \tau(X_3') > \ \ldots > 0$.

4. Given a monetary budget *C* and a unit cost c_i, find N_1^*:

$$\sum_{i=1}^{N_1^*} c_i = C$$

Optimal constrained assignment

- Eligibility, budget, ethical, or institutional constrains make policymakers unable to implement the *optimal unconstrained policy assignment*

- They are obliged to rely on a constrained assignment rule selecting treated units according to their characteristics

- The welfare thus obtained may drop down

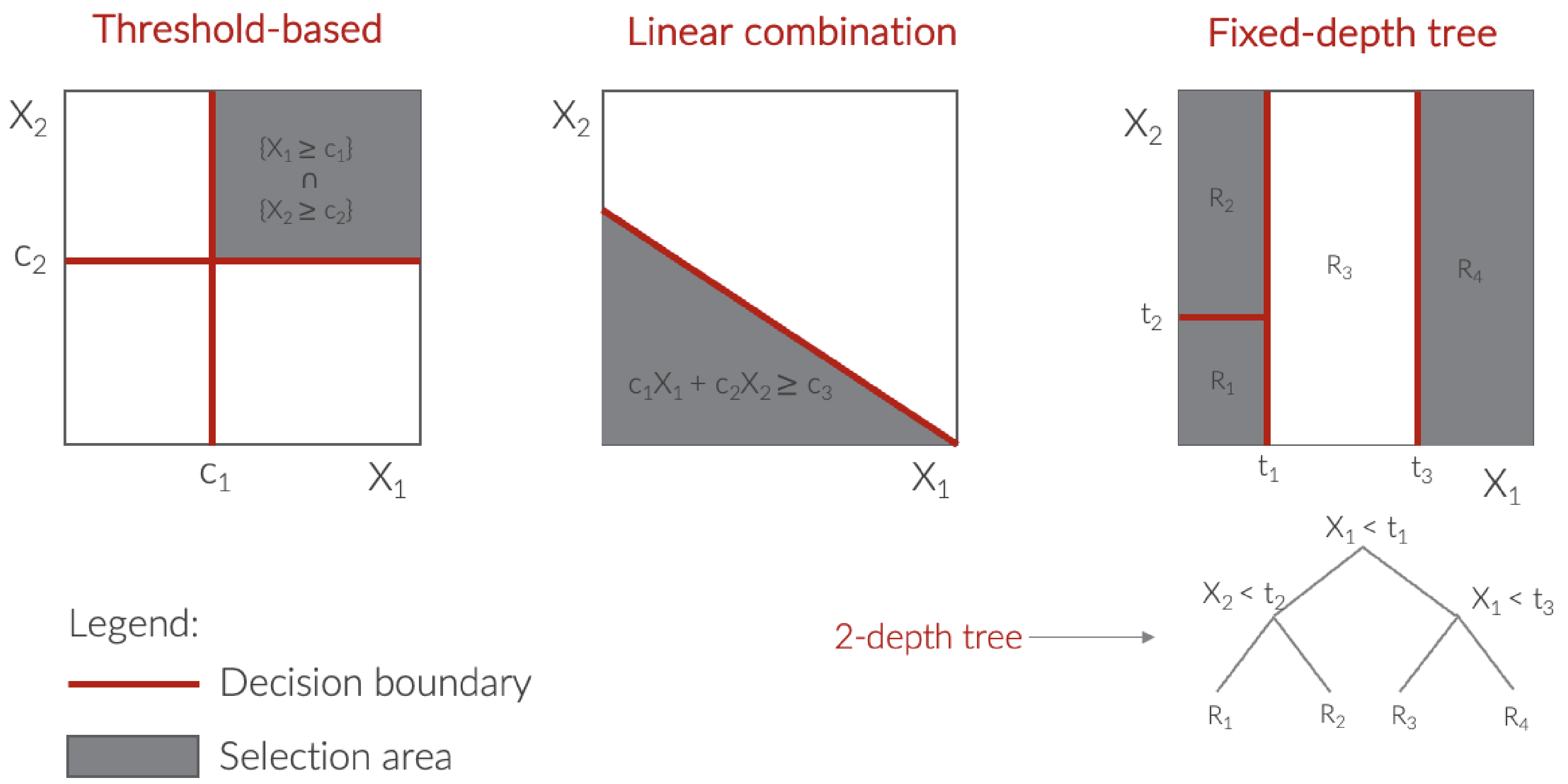
- Policymakers can however produce the largest feasible constrained welfare

Policy classes

There exist however several **classes of policies** used by policymakers to select in a constrained decision context. The most popular are:

- Threshold-based
- Linear combination
- Fixed-depth decision trees

Policy **classes**
(decision boundaries)



Threshold-based policy

OPTIMAL CONSTRAINED TREATMENT RULE

Splitting feature

Unit selection function

$$\boxed{\hat{T}_i(x, c_x)} = \hat{T}_i^* \cdot \boxed{\mathbf{1}[x >= c_x]}$$

Optimal unconstrained policy

$\hat{T}_i^* = \mathbf{1}[\hat{\tau}(X_i) > 0]$

Threshold value

Computing the *optimal thresholds*

OPTIMAL CONSTRAINED WELFARE

The corresponding welfare is a function of c_x:

$$\widehat{W}(x, c_x) = \sum_{i=1}^{N} \hat{T}_i(x, c_x) \cdot \hat{\tau}(X_i)$$

We define the optimal choice of the threshold c_x as the one maximizing $\widehat{W}(x, c_x)$ over c_x:

$$c_x^* = \texttt{argmax}_{c_x} [\widehat{W}(x, c_x)]$$

If c_x^* exists, the estimated optimal constrained welfare will thus be equal to $\widehat{W}(c_x^*)$.

Multiple selection variables

OPTIMAL CONSTRAINED TREATMENT RULE

Policymakers rely on two or more selection indicators

Splitting feature x

Splitting feature z

$$\hat{T}_i(c_x, c_z) = \hat{T}_i^* \cdot \mathbf{1}[x >= c_x] \cdot \mathbf{1}[z >= c_z]$$

Optimal unconstrained policy

Threshold Value for x

Threshold Value for z

Estimation

Procedure. Threshold–based optimal policy assignment

1. Suppose to have data from an RCT or from an observational study consisting of the information triple (Y, X, T) available for every unit involved in the program.
2. Run a quasi–experimental method with observable heterogeneity, estimate $\tau(X)$, and compute the (estimated) actual total welfare of the policy $\widehat{W}$.
3. Identify the estimated optimal unconstrained policy $\hat{T}^*$, and compute $\widehat{W}^*$, i.e. the estimated maximum total welfare achievable by the policy, and estimate the regret as $\widehat{W}^* - \widehat{W}$.
4. Consider an estimated constrained selection rule $\hat{T}(x, c)$ based on a given set of selection variables, x, and related thresholds, c, and define the estimated maximum constrained welfare as $\widehat{W}(x, c)$.
5. Build a greed of K possible values for $c \in \{c_1, ..., c_K\}$, compute the optimal vector of thresholds c_{k^*} and the corresponding maximum estimated welfare $\widehat{W}(x, c_{k^*})$ thus achieved.

Linear-combination policy

3 parameters

Generates a **score** to compare with a threshold

$$\hat{T}_i(c_1, c_2, c_3) = \hat{T}_i^* \cdot \mathbf{1}[\underbrace{c_1 x_1 + c_2 x_2}_{\text{score}} \geq \underbrace{c_3}_{\text{threshold}}]$$

Optimal unconstrained policy

Decision-tree policy

Fixed-depth decision tree. Within this policy class, given two selection variables x_1 and x_2, and three thresholds c_1, c_2, and c_3, the estimated assignment to treatment is:

$$\widehat{T}_i(z(1), z(2), z(3), c_1, c_2, c_3) = \widehat{T}_i^* \cdot \{\mathbf{1}[z(1) >= c_1] \cdot \mathbf{1}[z(2) >= c_2] + (1 - \mathbf{1}[z(1) >= c_1]) \cdot \mathbf{1}[z(3) >= c_3]\}$$

where each $z(j)$ – with $j = 1, 2, 3$ – can be either x_1 and x_2.

The corresponding welfare is given by:

$$\widehat{W}(x, \,.\,) = \sum_{i=1}^{N} \widehat{T}_i(z(1), z(2), z(3), c_1, c_2, c_3) \cdot \widehat{\tau}(X_i)$$

SOFTWARE

We formed a research group for **OPL software implementation** within the PNRR project **FOSSR**:

Stata

Cerulli (CNR), `opl` command

R

Guardabascio (Perugia University) and Brogi (Istat)

Python

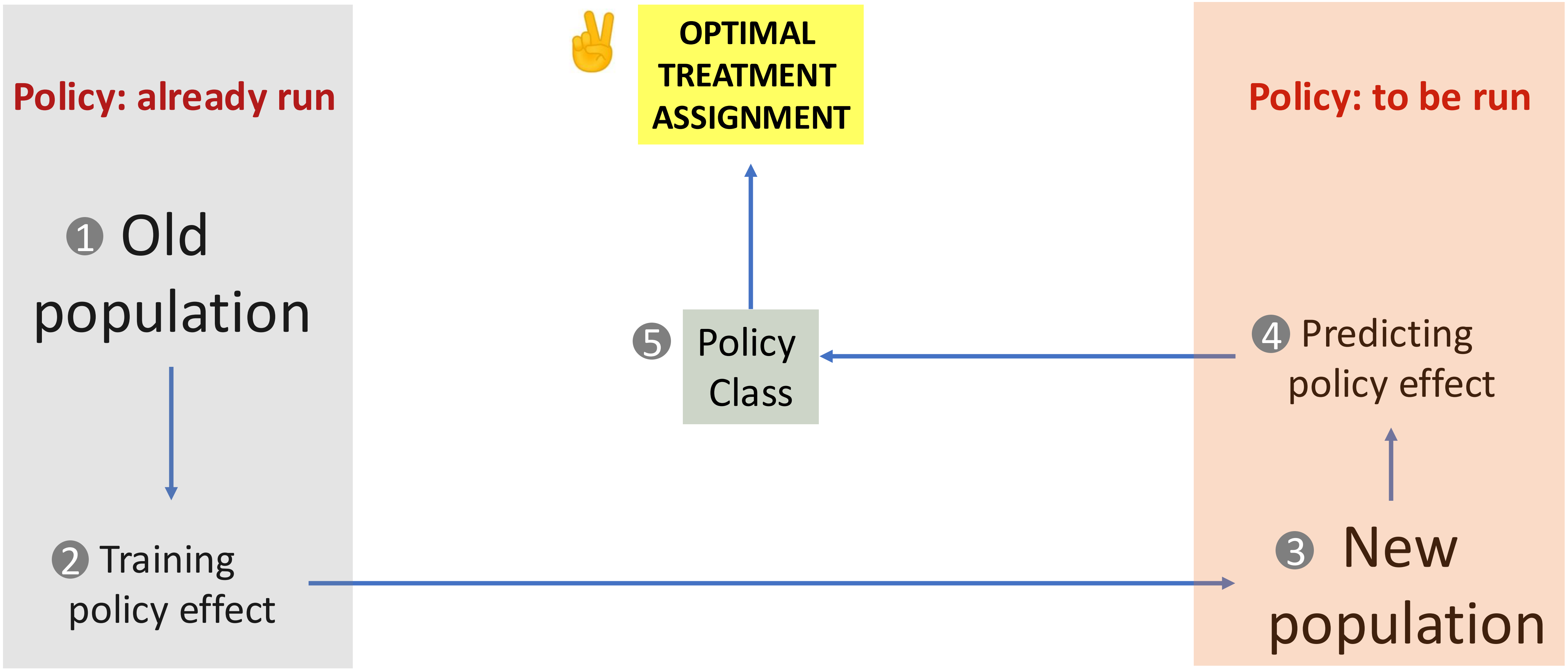
De Fausti (Istat)

OPL — Stata package for optimal policy learning

Syntax

command ... [, *options*]

command	Description
make_cate	Estimation of the conditional average treatment effect (CATE)
opl_tb	Threshold-based optimal policy learning
opl_tb_c	Threshold-based policy learning at specific threshold values
opl_lc	Linear-combination optimal policy learning
opl_lc_c	Linear-combination policy learning at specific parameters' values
opl_dt	Decision-tree optimal policy learning
opl_dt_c	Decision-tree policy learning at specific splitting variables and threshold values


Policy: already run
❶ Old population
❷ Training policy effect
OPTIMAL TREATMENT ASSIGNMENT
❺ Policy Class
Policy: to be run
❹ Predicting policy effect
❸ New population

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Consiglio Nazionale delle Ricerche

Treatment assignment in the ***old*** population

Predicted effect on the ***new*** population

```
make_cate outcome features , treatment(varname) new_cate(name) train_cate(name) new_data(name)
        [model(model_type) type(algorithm_type)]
```

Model depends on the **nature** of the outcome

Algorithm to estimate **CATE**

Predicted effect on the ***old*** population

Data on the ***new*** population

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Consiglio Nazionale delle Ricerche

THRESHOLD-BASED POLICY

opl_tb — Threshold-based optimal policy learning

Syntax

opl_tb , xlist(*var1 var2***) cate(***varname***)**

Description

opl_tb is a command implementing optimal ex-ante treatment assignment using as policy class a threshold-based (or quadrant) approach.

opl_tb_c — Threshold-based policy learning at specific threshold values

Syntax

opl_tb_c , xlist(*var1 var2***) cate(***varname***) c1(***number***) c2(***number***)** [**graph**]

Description

opl_tb_c is a command implementing ex-ante treatment assignment using as policy class a threshold-based (or quadrant) approach at specific threshold values *c1* and *c2* for respectively the selection variables *var1* and *var2*.

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Consiglio Nazionale delle Ricerche

LINEAR-COMBINATION POLICY

opl_lc — Linear-combination optimal policy learning

Syntax

opl_lc , **xlist(***var1 var2***) cate(***varname***)**

Description

opl_lc is a command implementing optimal ex-ante treatment assignment using as policy class a linear-combination of variables *var1* and *var2*: c1*var1+c2*var2=c3.

opl_lc_c —

Linear-combination policy learning at specific parameters' values

Syntax

opl_lc_c , **xlist(***var1 var2***) cate(***varname***) c1(***number***) c2(***number***)** [**graph**]

Description

opl_lc_c is a command implementing ex-ante treatment assignment using as policy class a linear-combination approach at specific parameters' values *c1*, *c2*, and *c3* for the linear-combination of variables *var1* and *var2*: c1*var1+c2*var2=c3.

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Consiglio Nazionale delle Ricerche

DECISION-TREE POLICY

opl_dt — Decision-tree optimal policy learning

Syntax

opl_dt , xlist(*var1 var2***) cate(***varname***)**

Description

opl_dt is a command implementing optimal ex-ante treatment assignment using as policy class a fixed-depth (1-layer) decision-tree based on selection variables *var1* and *var2*.

opl_dt_c — Decision-tree policy learning at specific splitting variables and threshold values

Syntax

opl_dt_c , xlist(*var1 var2***) cate(***varname***) c1(***number***) c2(***number***)** [**graph**]

Description

opl_dt_c is a command implementing ex-ante treatment assignment using as policy class a fixed-depth (1-layer) decision-tree at specific splitting variables and threshold values.

Application

- **DATA**: National Supported Work Demonstration (NSWD), an RCT by LaLonde (1986).
- **TARGET**: Effect of a 1976 job training program on people real earnings in 1978
- **CONTROLS**: age, race, educational attainment, previous employment condition, real earnings in 74 and 75

Application 1

opl_tb_c

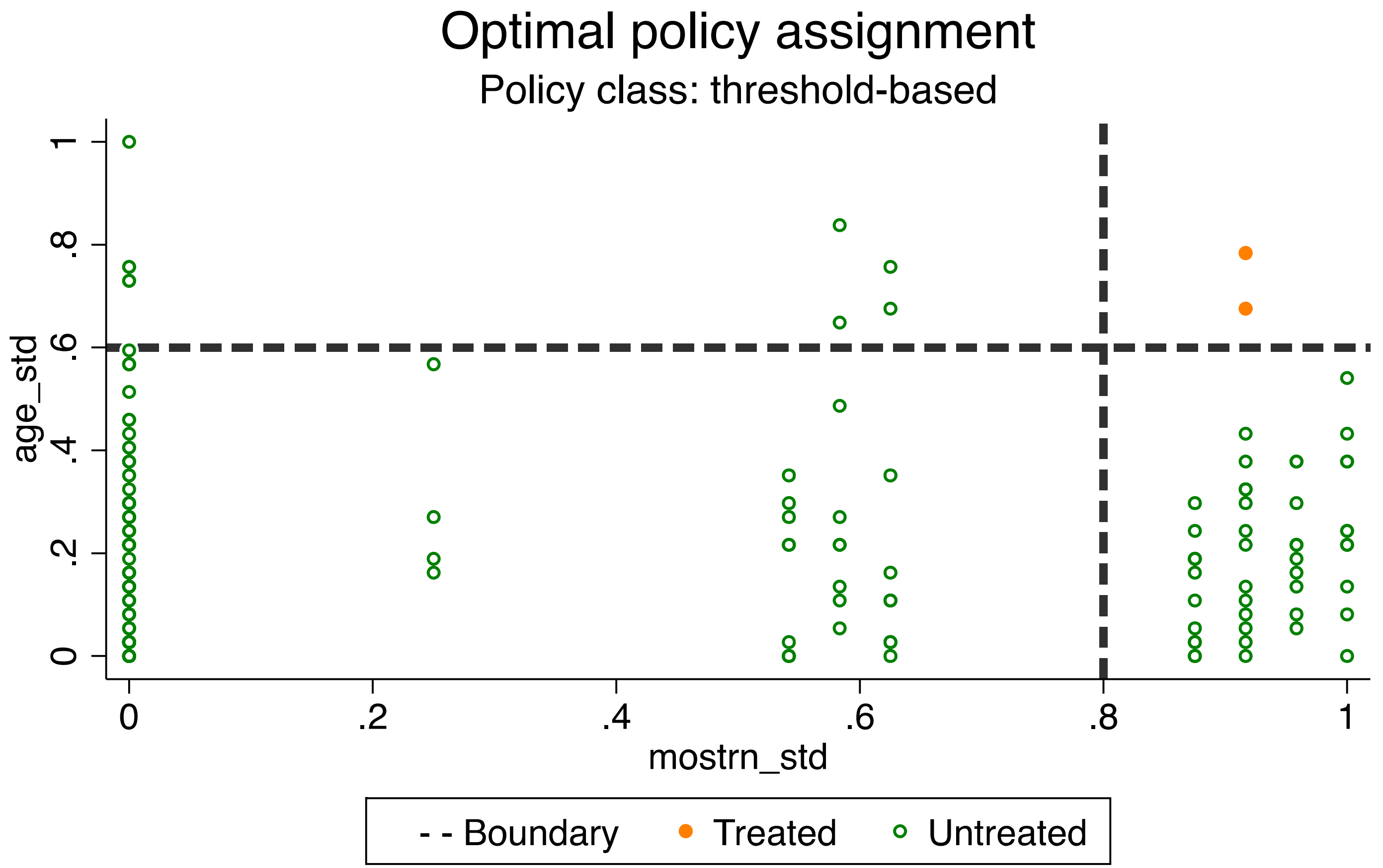
```
Load initial dataset
    sysuse JTRAIN2, clear
Split the original data into a "old" (training) and "new" (testing) dataset
    get_train_test, dataname(jtrain) split(0.60 0.40) split_var(svar) rseed(101)
Use the "old" dataset (i.e. policy) for training
    use jtrain_train , clear
Set the outcome
    global y "re78"
Set the features
    global x "re74 re75 age agesq nodegree"
Set the treatment variable
    global w "train"
Set the selection variables
    global z "age mostrn"
Run "make_cate" and generate training (old policy) and testing (new policy) CATE predictions
    make_cate $y $x , treatment($w) model("ra") new_cate("my_cate_new") train_cate("my_cate_train") new_data("jtrain_test")
Generate a global macro containing the name of the variable "cate_new"
    global T `e(cate_new)'
Select only the "new data"
    keep if _train_new_index=="new"
Drop "my_cate_train" as in the new dataset treatment assignment and outcome performance are unknown
    drop my_cate_train $w $y
Run "opl_tb" to find the optimal thresholds
    opl_tb , xlist($z) cate($T)
Save the optimal threshold values into two global macros
    global c1_opt=e(best_c1)
    global c2_opt=e(best_c2)
Run "opl_tb_c" at optimal thresholds and generate the graph
    opl_tb_c , xlist($z) cate($T) c1($c1_opt) c2($c2_opt) graph
Tabulate the variable "_units_to_be_treated"
    tab _units_to_be_treated , mis
```

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Consiglio Nazionale delle Ricerche

```
Policy class: Threshold-based

Main results

Learner = Regression adjustment              Target variable =
N. of units = 178                            Selection variables = age mostrn
Threshold value c1 = .60000002               Threshold value c2 = .79999999
Average unconstrained welfare = 2.0673337    Average constrained welfare = 2.885844
Percentage of treated = 1.1                  N. of treated = 2
N. of untreated = 176
```

```
. tab _units_to_be_treated , mis
```

1 = unit to treat; 0 = unit not to treat	Freq.	Percent	Cum.
0	176	98.88	98.88
1	2	1.12	100.00
Total	178	100.00	



Expected unconstrained average welfare = 2.07
Expected constrained average welfare = 2.89
Percentage of treated units = 1.1%

Finanziato dall'Unione europea
NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA
Consiglio Nazionale delle Ricerche

Application 2
opl_lc_c

```
Load initial dataset
    sysuse JTRAIN2, clear
Split the original data into a "old" (training) and "new" (testing) dataset
    get_train_test, dataname(jtrain) split(0.60 0.40) split_var(svar) rseed(101)
Use the "old" dataset (i.e. policy) for training
    use jtrain_train , clear
Set the outcome
    global y "re78"
Set the features
    global x "re74 re75 age agesq nodegree"
Set the treatment variable
    global w "train"
Set the selection variables
    global z "age mostrn"
Run "make_cate" and generate training (old policy) and testing (new policy) CATE predictions
    make_cate $y $x , treatment($w) model("ra") new_cate("my_cate_new") train_cate("my_cate_train") new_data("jtrain_test")
Generate a global macro containing the name of the variable "cate_new"
    global T `e(cate_new)'
Select only the "new data"
    keep if _train_new_index=="new"
Drop "my_cate_train" as in the new dataset treatment assignment and outcome performance are unknown
    drop my_cate_train $w $y
Run "opl_lc" to find the optimal linear-combination parameters
    opl_lc , xlist($z) cate($T)
Save the optimal linear-combination parameters into three global macros
    global c1_opt=e(best_c1)
    global c2_opt=e(best_c2)
    global c3_opt=e(best_c3)
Run "opl_lc_c" at optimal linear-combination parameters and generate the graph
    opl_lc_c , xlist($z) cate($T) c1($c1_opt) c2($c2_opt) c3($c3_opt) graph
Tabulate the variable "_units_to_be_treated"
    tab _units_to_be_treated , mis
```

```
Policy class: Linear-combination

Main results

Learner = Regression adjustment              Target variable =
N. of units = 178                            Selection variables = age mostrn
Lin. comb.parameter c1 = .59999999           Lin. comb.parameter c2 = .45000001
Lin. comb.parameter c3 = .8                  Average unconstrained welfare = 2.0673337
Average constrained welfare = 2.885844       Percentage of treated = 1.1
N. of treated = 2                            N. of untreated = 176
```

```
. tab _units_to_be_treated , mis
```

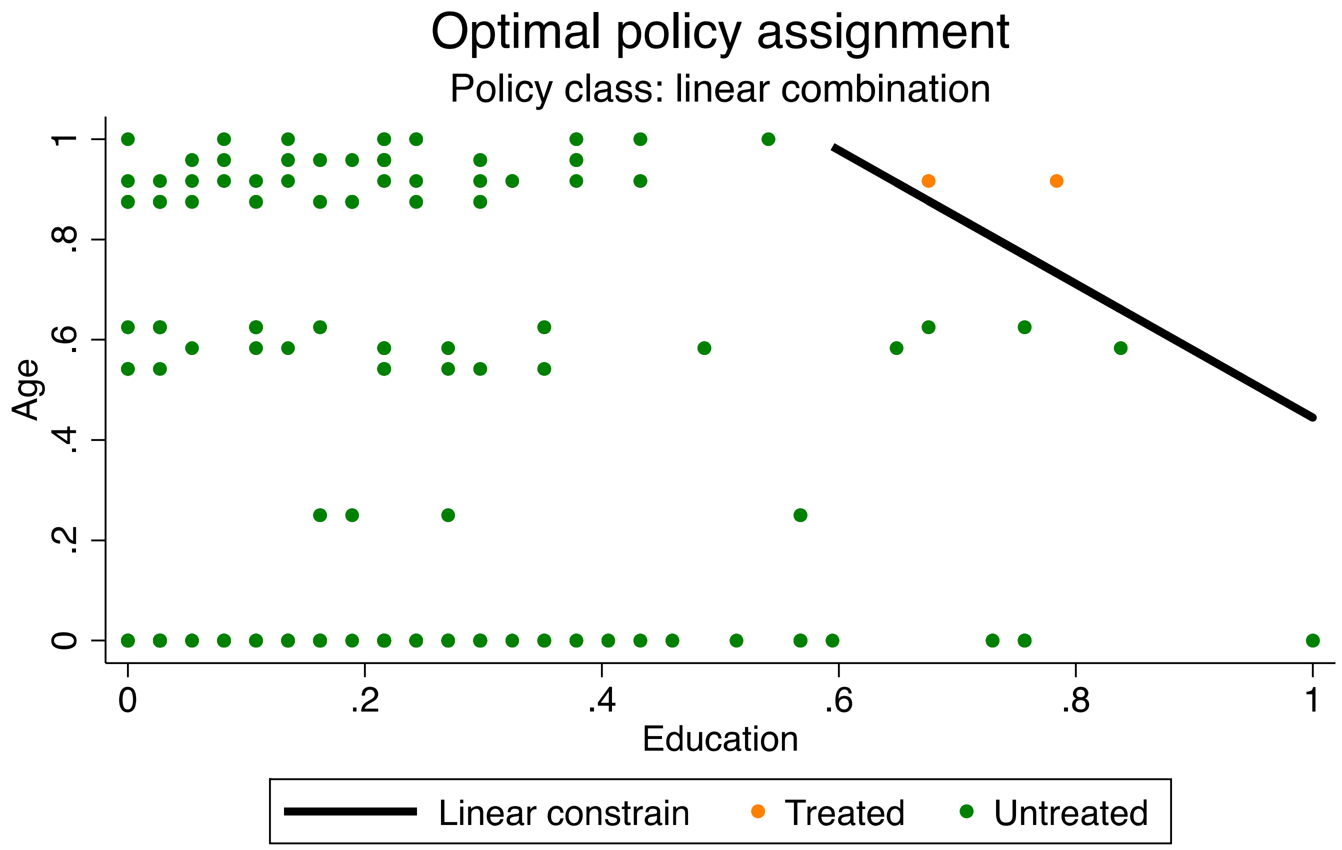
1 = unit to treat; 0 = unit not to treat	Freq.	Percent	Cum.
0	176	98.88	98.88
1	2	1.12	100.00
Total	178	100.00	

Finanziato dall'Unione europea NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA

Consiglio Nazionale delle Ricerche



Expected unconstrained average welfare = 2.07
Expected constrained average welfare = 2.89
Percentage of treated units = 1.1%

Finanziato dall'Unione europea
NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA
Consiglio Nazionale delle Ricerche

Application 3

opl_dt_c

```
Load initial dataset
    sysuse JTRAIN2, clear
Split the original data into a "old" (training) and "new" (testing) dataset
    get_train_test, dataname(jtrain) split(0.60 0.40) split_var(svar) rseed(101)
Use the "old" dataset (i.e. policy) for training
    use jtrain_train , clear
Set the outcome
    global y "re78"
Set the features
    global x "re74 re75 age agesq nodegree"
Set the treatment variable
    global w "train"
Set the selection variables
    global z "age mostrn"
Run "make_cate" and generate training (old policy) and testing (new policy) CATE predictions
    make_cate $y $x , treatment($w) model("ra") new_cate("my_cate_new") train_cate("my_cate_train") new_data("jtrain_test")
Generate a global macro containing the name of the variable "cate_new"
    global T `e(cate_new)'
Select only the "new data"
    keep if _train_new_index=="new"
Drop "my_cate_train" as in the new dataset treatment assignment and outcome performance are unknown
    drop my_cate_train $w $y
Run "opl_dt" to find the optimal linear-combination parameters
    opl_dt , xlist($z) cate($T)
Save the optimal splitting variables into three global macros
    global x1_opt `e(best_x1)'
    global x2_opt `e(best_x2)'
    global x3_opt `e(best_x3)'
Save the optimal splitting thresholds into three global macros
    global c1_opt=e(best_c1)
    global c2_opt=e(best_c2)
    global c3_opt=e(best_c3)
Run "opl_dt_c" at optimal splitting variables and corresponding thresholds and generate the graph
    opl_dt_c , xlist($z) cate($T) c1($c1_opt) c2($c2_opt) c3($c3_opt) x1($x1_opt) x2($x2_opt) x3($x3_opt) graph
Tabulate the variable "_units_to_be_treated"
    tab _units_to_be_treated , mis
```

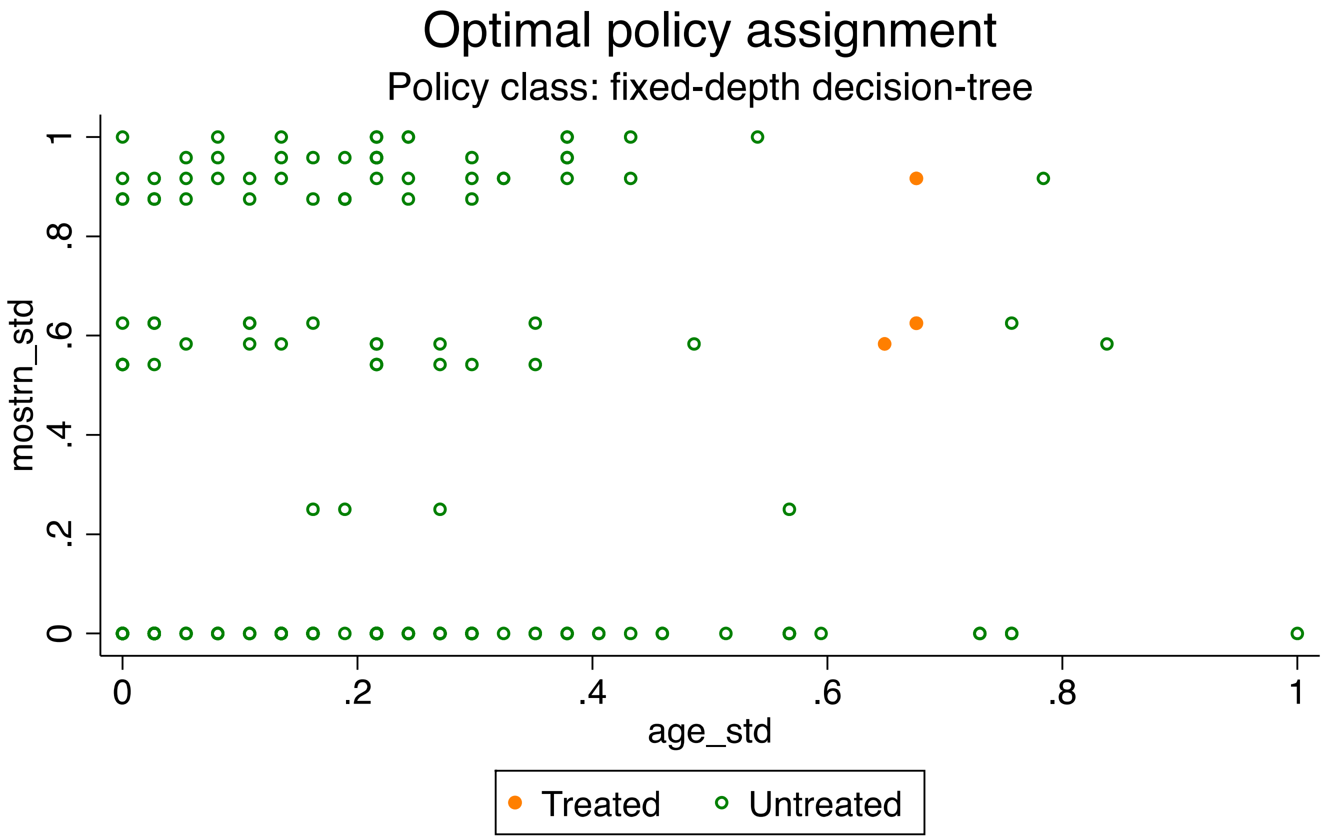
Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Consiglio Nazionale delle Ricerche

```
Policy class: Fixed-depth decision-tree

Main results

Learner = Regression adjustment                    Target variable =
N. of units = 178                                  Selection variables =
Threshold first splitting var. = .69999999         Threshold second splitting var. = .89999998
Threshold third splitting var. =  = .60000002      Average unconstrained welfare = 2.0673337
Average constrained welfare = 4.2417823            Percentage of treated = 1.7
N. of treated = 3                                  N. of untreated = 175
First splitting variable x1 = age                  Second splitting variable x2 = age
Third splitting variable x3 = age
```

```
. tab _units_to_be_treated , mis
```

1 = unit to treat; 0 = unit not to treat	Freq.	Percent	Cum.
0	175	98.31	98.31
1	3	1.69	100.00
Total	178	100.00	

Finanziato dall'Unione europea
NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA
Consiglio Nazionale delle Ricerche



Expected unconstrained average welfare = 2.07
Expected constrained average welfare = 4.24
Percentage of treated units = 1.7%

Finanziato dall'Unione europea
NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

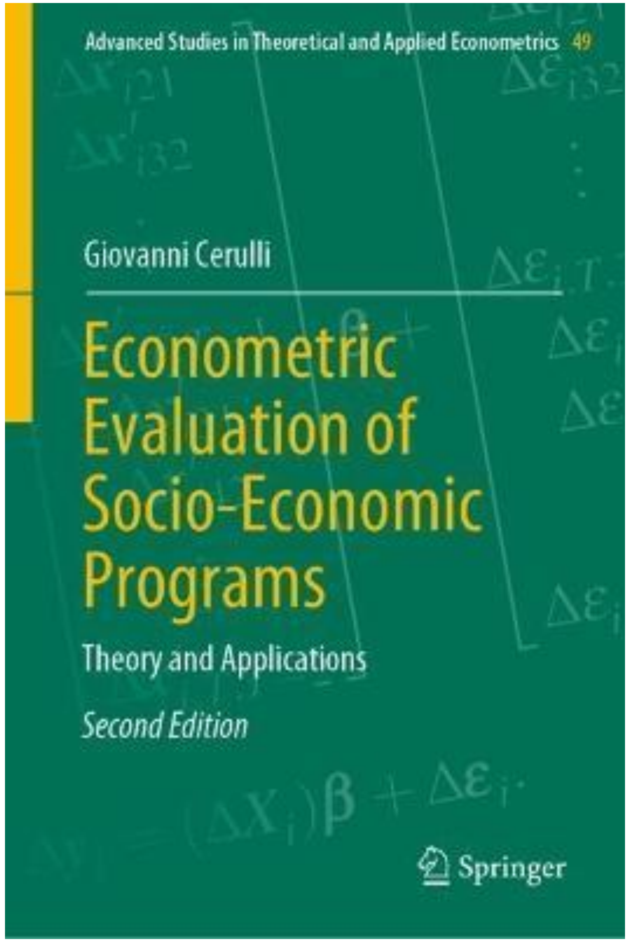
Consiglio Nazionale delle Ricerche

CONCLUSIONS

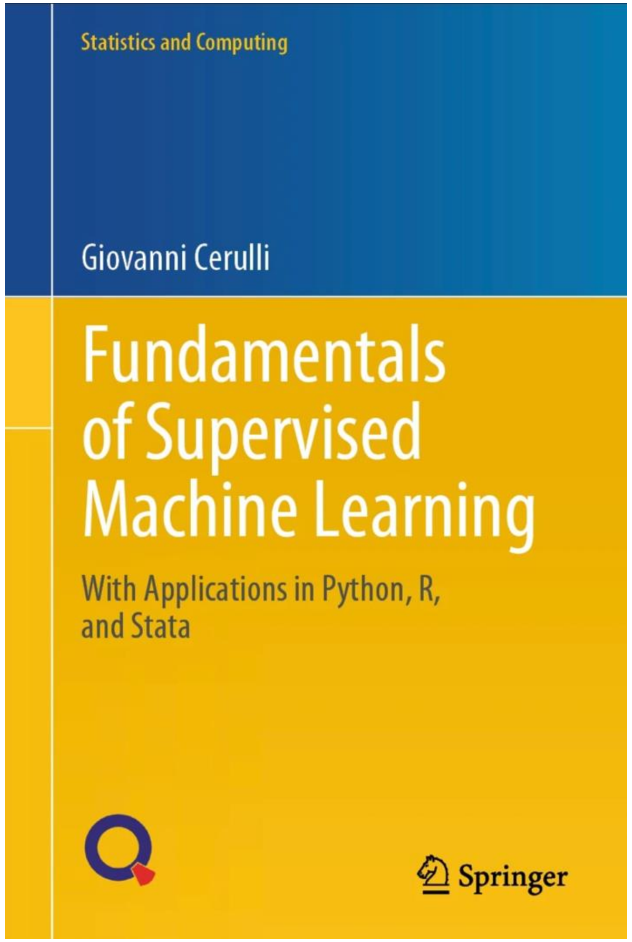
- Policy Learning: new frontier of econometrics of prog evaluation
- Theory-driven and data-driven approaches can complement
- Extensions to unobservable selection quite straightforward
- Machine Learning algorithms for estimating $\tau(X)$
- Welfare monotonicity and data sparseness major problems
- Monotonicity solved by "menu strategy"
- Generalization to other policy classes
- Providing Stata/R/Python software implementation

Books for learning about Causal Inference and Machine Learning

Causal inference



Machine learning



Finanziato dall'Unione europea
NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA
Consiglio Nazionale delle Ricerche

THANK YOU!

fossr.dissemination@ircres.cnr.it

@fossrproject

fossr.eu

fossr-eu

@fossr

zenodo.org/communities/fossr