Consiglio Nazionale delle Ricerche

IRCrES

ISTITUTO di RICERCA sulla CRESCITA ECONOMICA SOSTENIBILE
INSTITUTE of RESEARCH on ECONOMIC SUSTAINABLE GROWTH

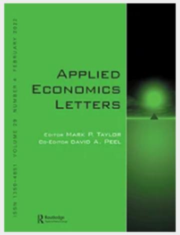
Optimal policy learning using Stata

Giovanni Cerulli

IRCrES-CNR

2023 Stata Conference
STANFORD, CALIFORNIA · 20–21 JULY

THIS PAPER



Applied Economics Letters >

Latest Articles

Submit an article | Journal homepage

Enter keywords, authors, DOI, ORCID e

Research Article

Optimal treatment assignment of a threshold-based policy: empirical protocol and related issues

Giovanni Cerulli

Published online: 30 Jan 2022

Download citation https://doi.org/10.1080/13504851.2022.2032577

INTRODUCTION

1. This paper deals with **ex-ante data-driven optimal design of (micro) policies**

2. It is embedded within the **optimal policy learning (OPL)** literature

3. It contributes by stressing the **policymaker perspective**

4. It suggests a **menu strategy** to deal with optimal solution's *monotonicity*

OPTIMAL POLICY LEARNING - 1

Optimal policy learning

Frontier of the “econometrics of program evaluation”

Changing policy perspective

From policy “ex-post” evaluation to “ex-ante” optimal policy design

Prediction based

Compared to ex-post evaluation (based on inference), OPL targets optimal “prediction”, entailing a central role of “machine learning”

DEFINITION OF OPL

What is policy learning?

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

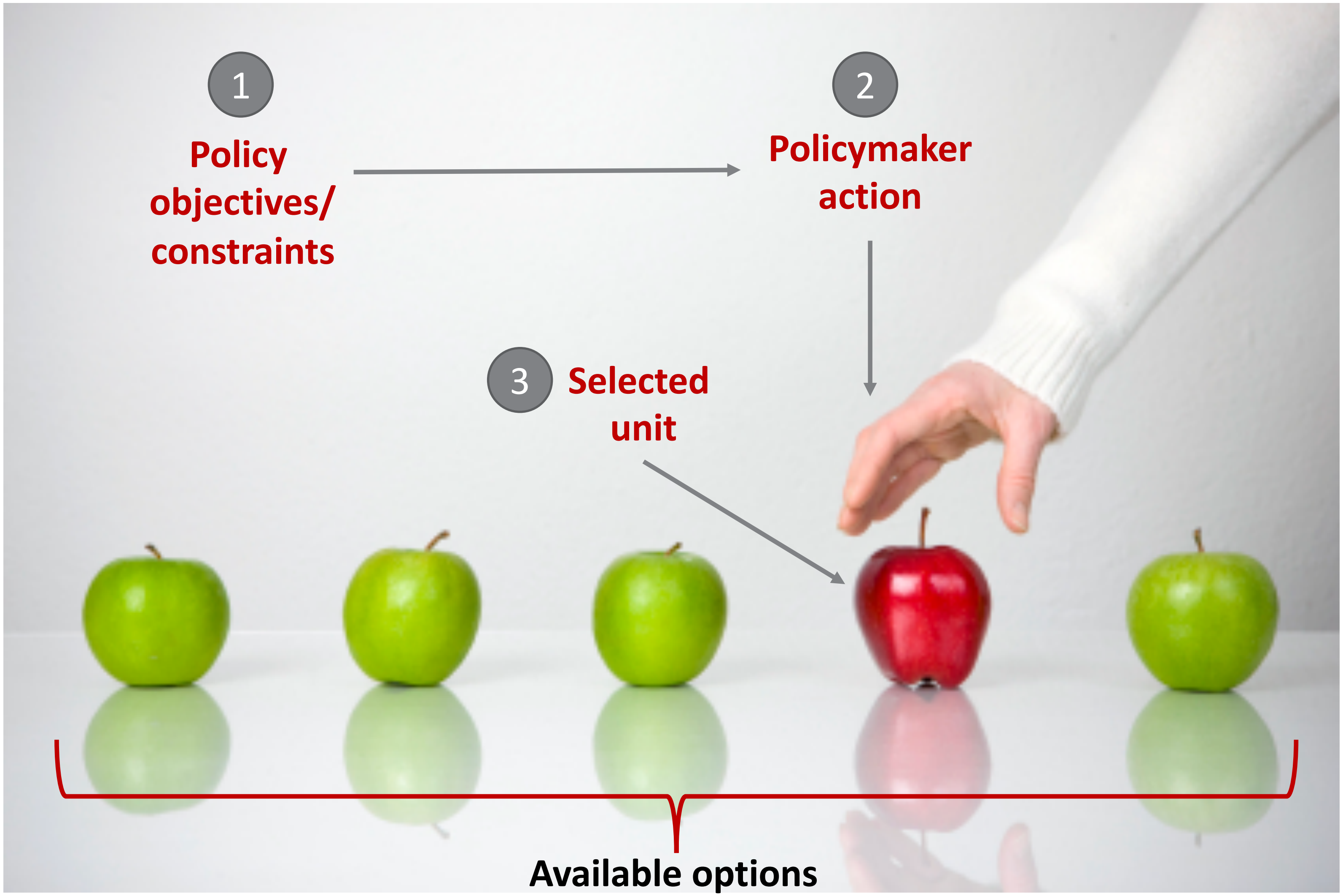
Optimal treatment assignment

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

POLICY AS A SELECTION PROBLEM



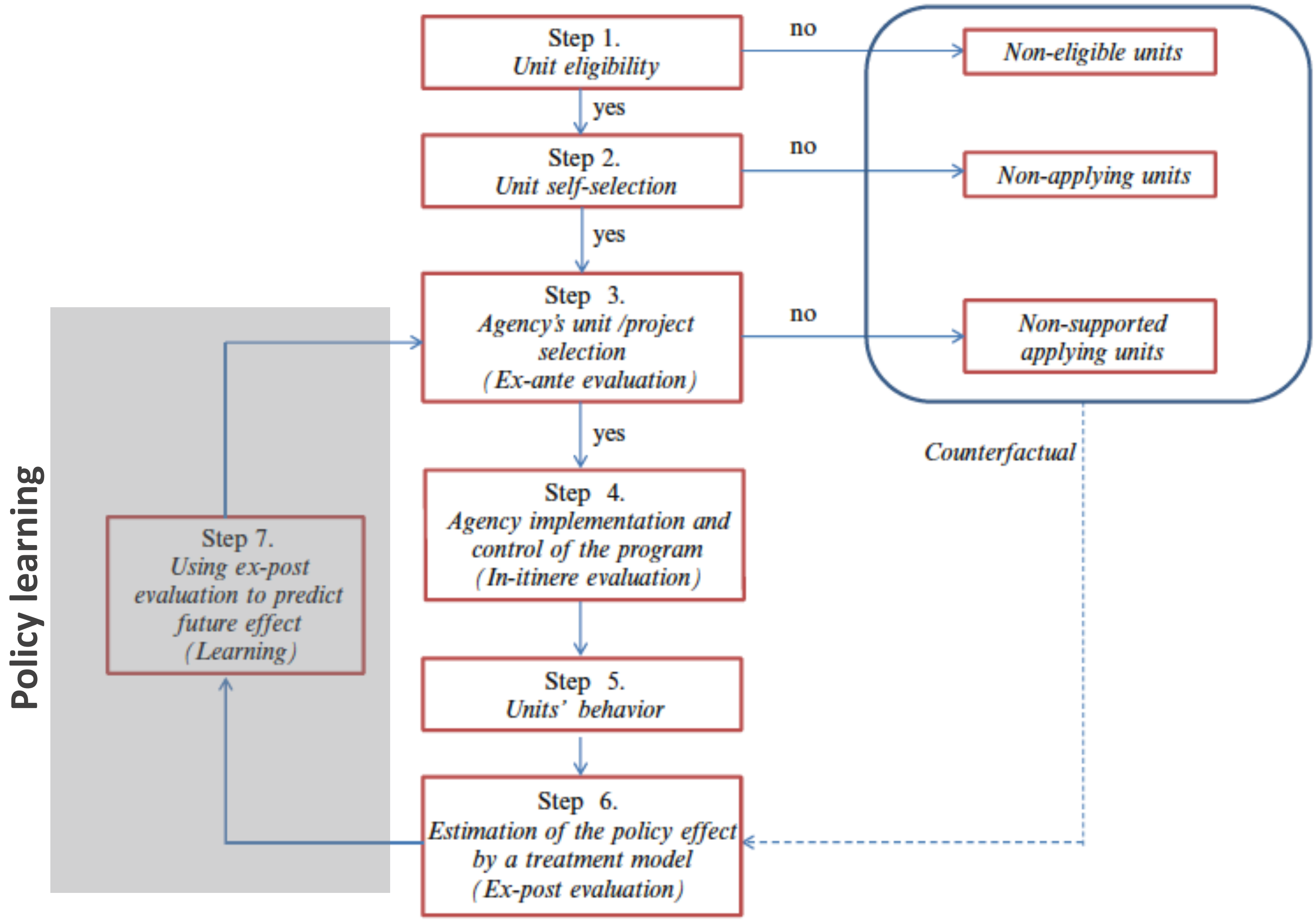
STATE-OF-THE ART - 1

Manski C.F. (2004), Statistical Treatment Rules for Heterogeneous Populations, *Econometrica*, 72, 4, 1221–1246.

Kitagawa T., Tetenov A. 2018. Who should be treated? empirical welfare maximization methods for treatment choice, *Econometrica*. 86, 2, 591–616.

Bhattacharya D., Dupas P. 2012. Inferring Welfare Maximizing Treatment Assignment under Budget Constraints. *Journal of Econometrics*, 167, 1, 168–196.

POLICY LEARNING WITHIN THE POLICY EVALUATION CYCLE



POLICY DIRECT AND INDIRECT EFFECT

Total
Policy effect

$$\gamma(S) = \alpha + \beta(S)$$

Direct
effect

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 - 1

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.

OPTIMAL TREATMENT ASSIGNMENT - 2

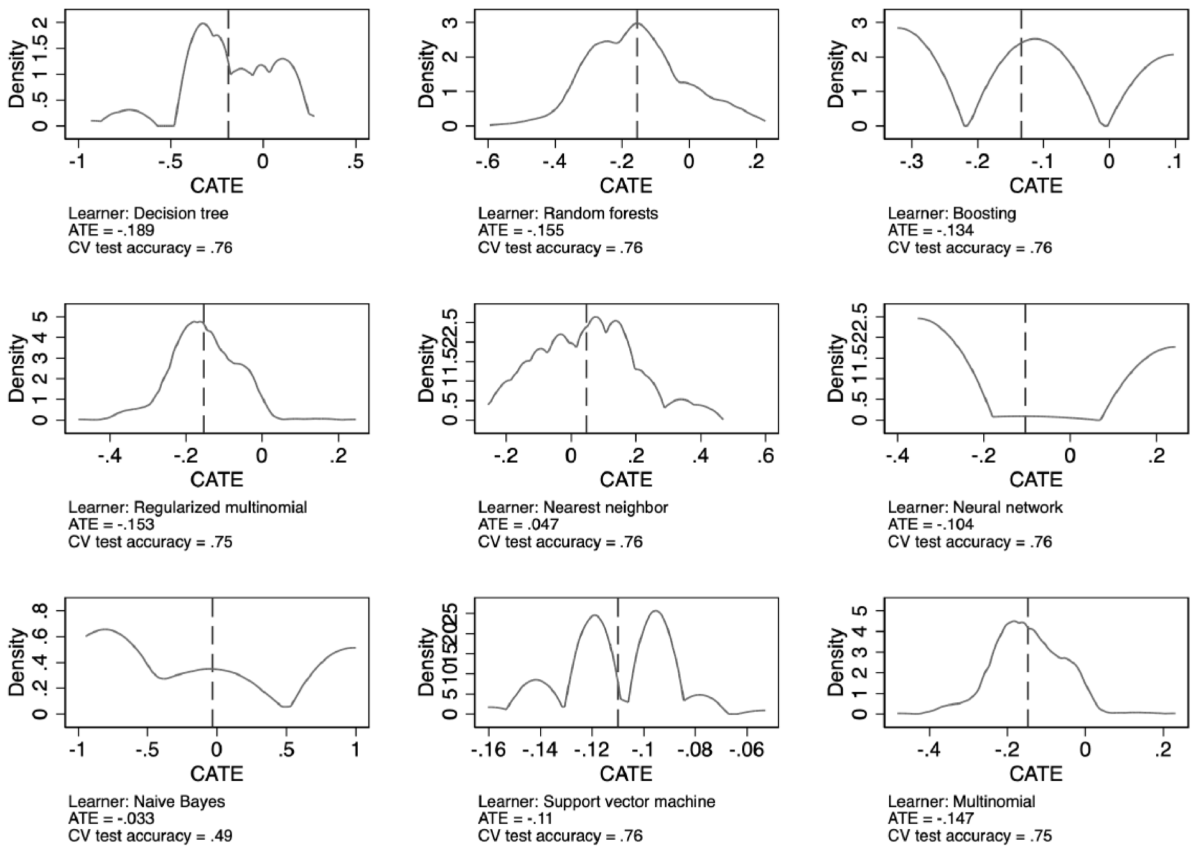
Under **selection-on-observables**, we know that:

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

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**).

OPTIMAL TREATMENT ASSIGNMENT - 3

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:

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

EXAMPLE

Example of an optimal policy assignment rule
The regret of this policy is equal to 16 = 26 - 10

ID	T	$\tau(X)$	$T \cdot \tau(X)$	T^*	$T^* \cdot \tau(X)$
1	1	9	9	1	9
2	1	-4	-4	0	0
3	1	5	5	1	5
4	0	6	0	1	6
5	0	-2	0	0	0
6	0	6	0	1	6
			10		26

Actual welfare reached

Maximum welfare feasible

regret **—> 26 – 10 = 16**

NAÏVE OPTIMAL SELECTION

1. Given {*X*,*Y*,*T*} from an already-implemented policy: estimate the **idiosyncratic effect $\tau(X)$**. 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

EXAMPLE OF CONSTRAINED ASSIGNMENT: UNIVARIATE THRESHOLD-BASED POLICY

- The policymaker wants to treat only "young" people
- In theory, he can continue to use the naïve approach, by excluding from treatment all the individuals with age smaller than a certain age $A*$
- The problem is that different $A*$ can induce different level of welfare
- The problem becomes that of **choosing $A*$ to maximize the effect/welfare**

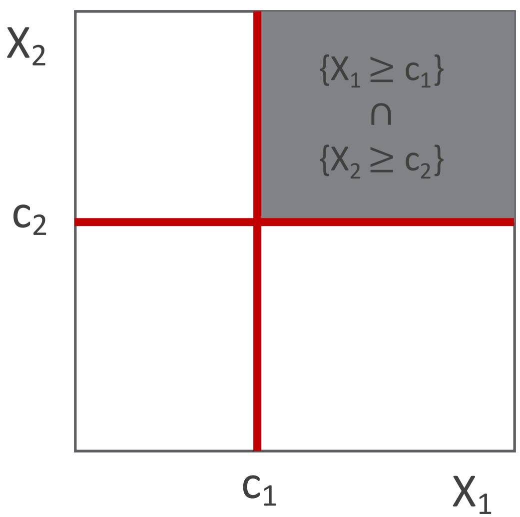
POLICY CLASSES

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

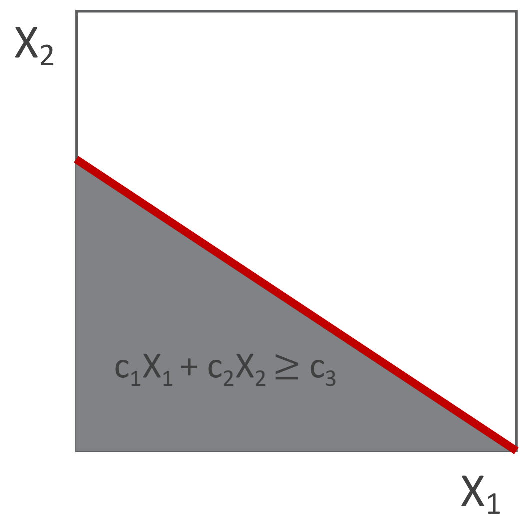
- ❑ Threshold-based
- ❑ Linear combination
- ❑ Fixed-depth decision trees

POLICY CLASSES (DECISION BOUNDARIES)

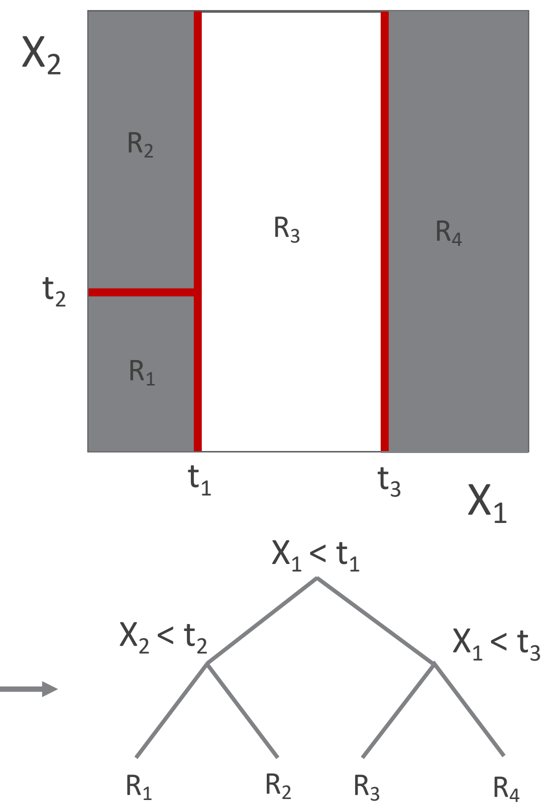
Threshold-based



Linear combination



Fixed-depth tree



2-depth tree →

Legend:

Decision boundary

Selection area

Threshold-based policy

OPTIMAL CONSTRAINED TREATMENT RULE

Splitting feature

Unit selection function

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

Optimal unconstrained policy

Threshold value

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

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 $\widehat{W}(c_x^*)$.

OPTIMAL CONSTRAINED TREATMENT RULE (*MULTIVARIATE CASE*)

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*

Consiglio Nazionale delle Ricerche
IRCrES
ISTITUTO di RICERCA sulla CRESCITA ECONOMICA SOSTENIBILE
INSTITUTE of RESEARCH on ECONOMIC SUSTAINABLE GROWTH

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 (BIVARIATE CASE)

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

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

score

threshold

Optimal unconstrained policy

APPLICATION

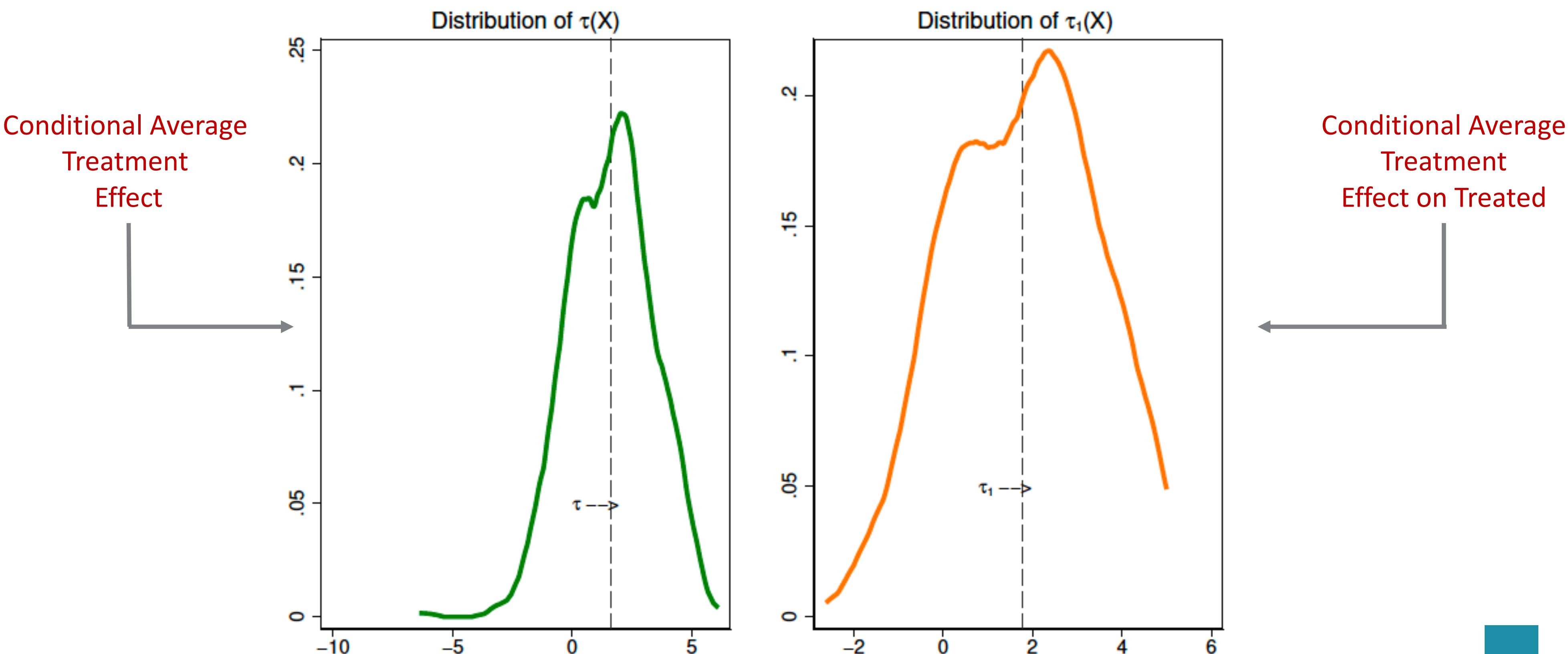
DATA: National Supported Work Demonstration (NSWD), an RCT by LaLonde (1986).

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

ESTIMATION OF ATE(X) AND ATET(X)

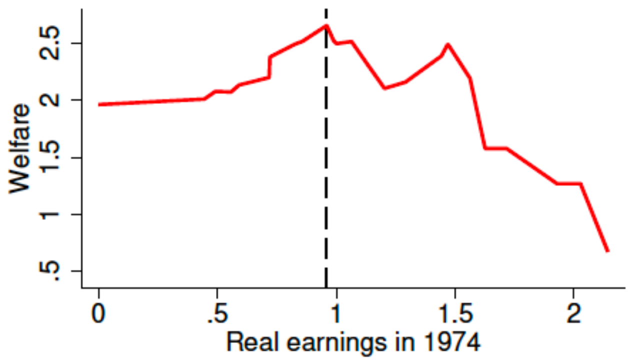
Figure 1: Distribution of $\hat{\tau}(X)$ and $\hat{\tau}_1(X)$. Program: National Supported Work Demonstration (NSWD). Data: LaLonde (1986). Target variable: Real earnings in 1978. Estimation technique: Regression–adjustment (with observable heterogeneity).



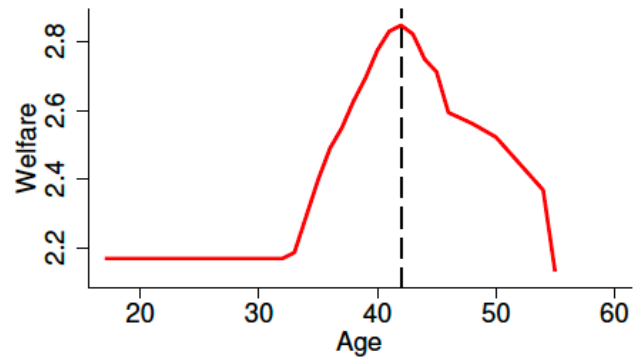
CONSTRAINED WELFARE MAXIMIZATION (UNIVARIATE)

Figure 2: Computation of the policy optimal selection threshold in univariate cases. Program: National Supported Work Demonstration (NSWD). Data: LaLonde (1986). Target variable: real earnings in 1978. Univariate selection variables: real earnings in 1974, age, and educational attainment.

AWG =
2.65 - 1.76 =
0.89



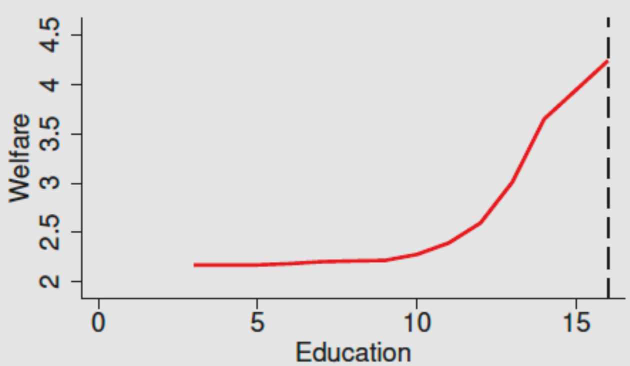
Optimal threshold = .96
Optimal average welfare = 2.65
Number of treated units = 108 out of 443



Optimal threshold = 42
Optimal average welfare = 2.85
Number of treated units = 16 out of 443

AWG =
2.85 - 1.76 =
1.09

AWG =
4.24 - 1.76 =
2.48



Optimal threshold = 16
Optimal average welfare = 4.24
Number of treated units = 0 out of 443

Monotonicity of welfare
on educational attainment

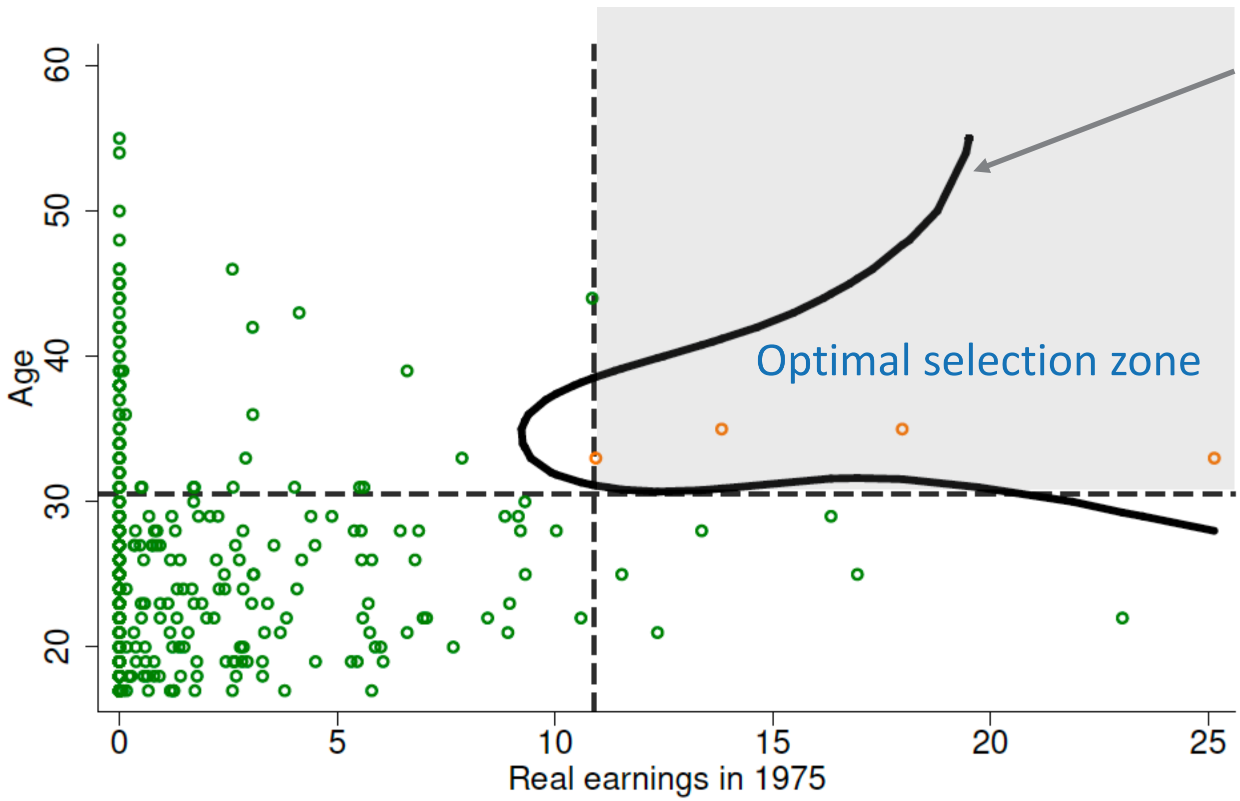
Reference ATET = 1.76
AWG = Average Welfare Gain

CONSTRAINED WELFARE MAXIMIZATION (BIVARIATE)

Reference ATET = 1.74

Average Welfare Gain = 3.99 – 1.74 = 2.255

Figure 3: Computation of the policy optimal decision boundary in the bivariate case. Program: National Supported Work Demonstration (NSWD). Data: LaLonde (1986). Target variable: real earnings in 1978. Bivariate selection variables: real earnings in 1975 and age.



Estimated Bayes optimal decision boundary

Average welfare = 3.995
Share of treated units = 1%
Optimal threshold for 'Age' = 30.5
Optimal threshold for 'Real earnings in 1975' = 10.9

EMPIRICAL WELFARE MAXIMIZATION:
RELEVANT ISSUES

1. Monotonicity

Welfare increases monotonically with a feature
=> *too few to treat or too many to treat*

2. Sparseness

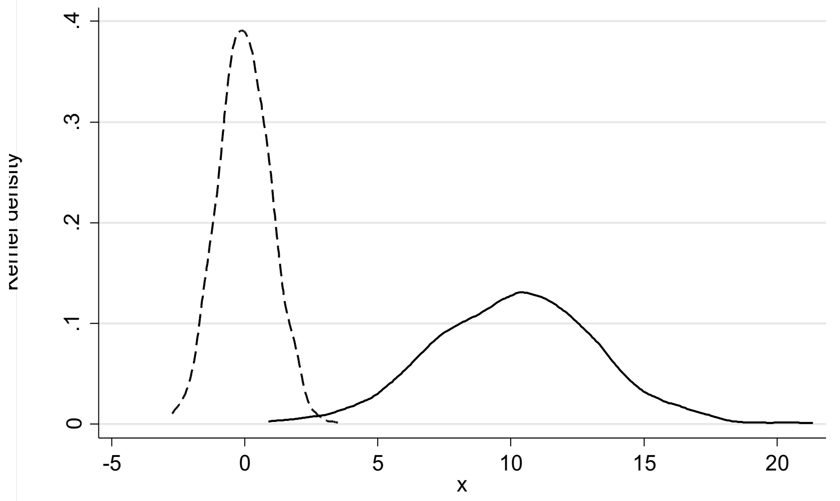
X' comes from a *different joint distribution* than *X*

Trade-offs arising in this case, so the best to do is offering the policymaker a **"menu"** of possible treatment choices given, for example, a pre-fixed budget

SPARSENESS

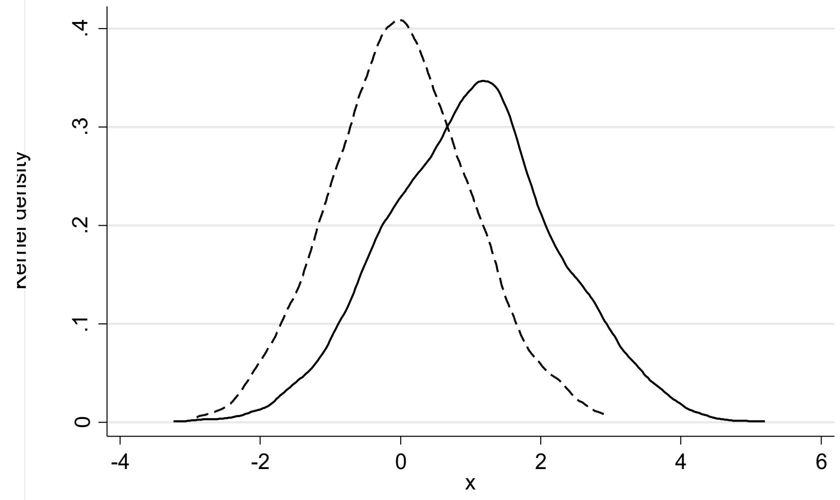
THE DISTRIBUTION OF *X* AND *X'* HAVE LOW OVERLAP

High sparseness



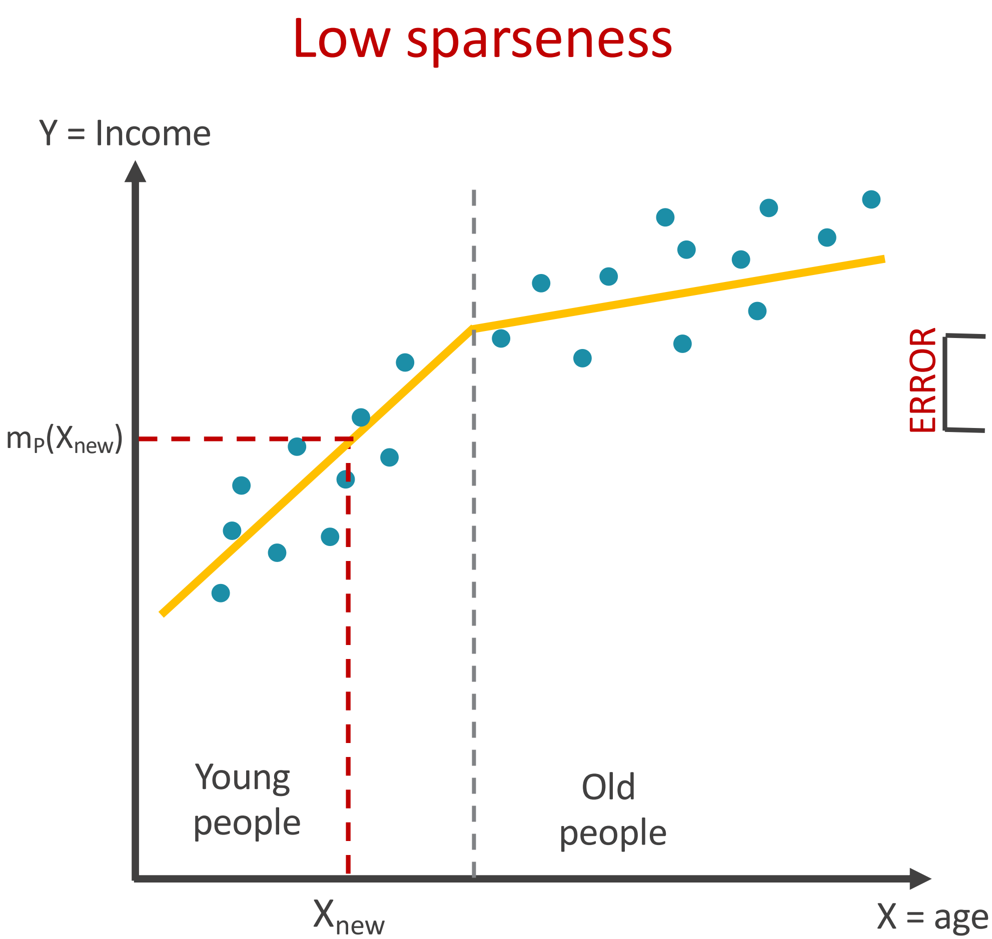
$$P(X) \neq P(X')$$

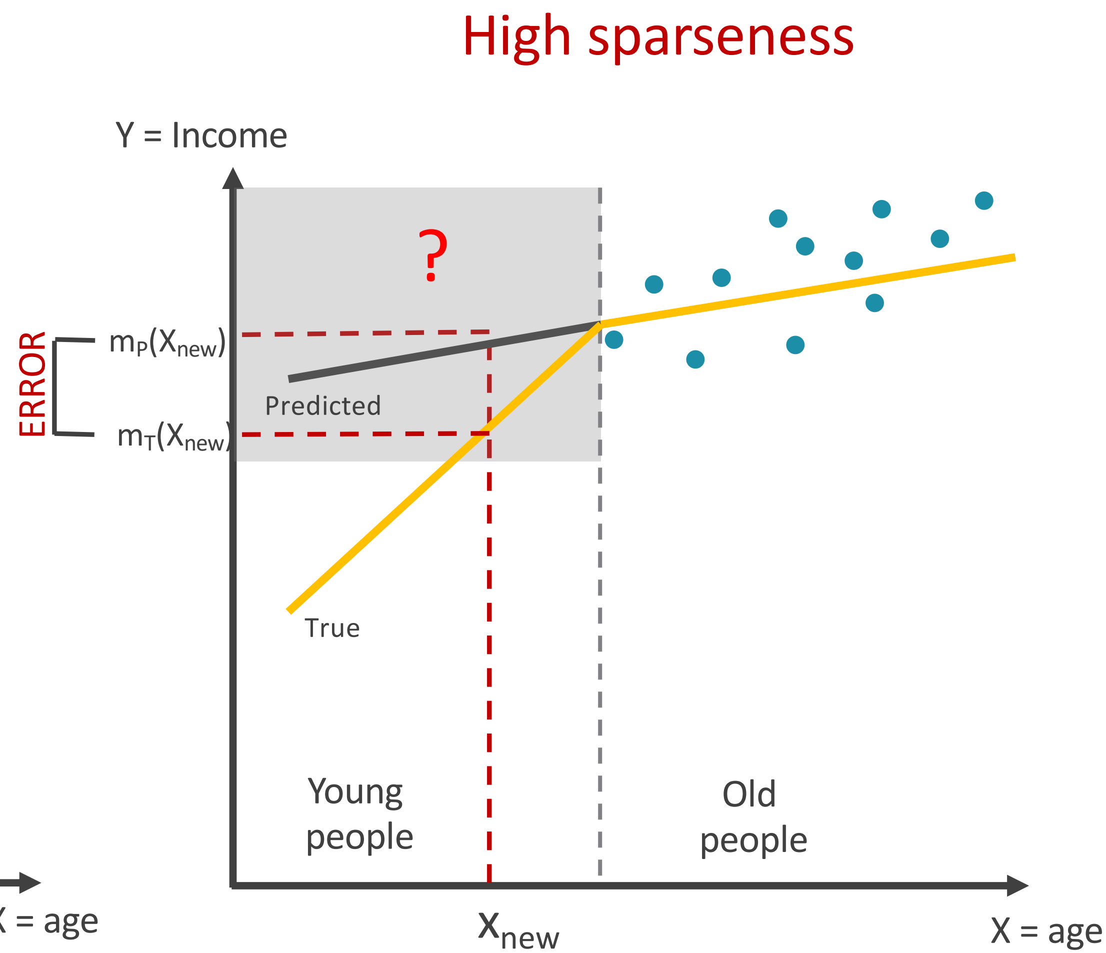
Low sparseness



$$P(X) \cong P(X')$$

Data sparseness **weakens** policy prediction





A SOLUTION TO MONOTONICITY

TRADE-OFFS AND THE "MENU-STRATEGY"

EXAMPLE

Computation of policy optimal decision boundaries in the bivariate case, when one of the two selection variables (age) is fixed at its optimal threshold, and the threshold of the other variable (education) is varying. Program: National Supported Work Demonstration (NSWD). Data: LaLonde (1986). Target variable: real earnings in 1978. Bivariate selection variables: age and educational attainment.

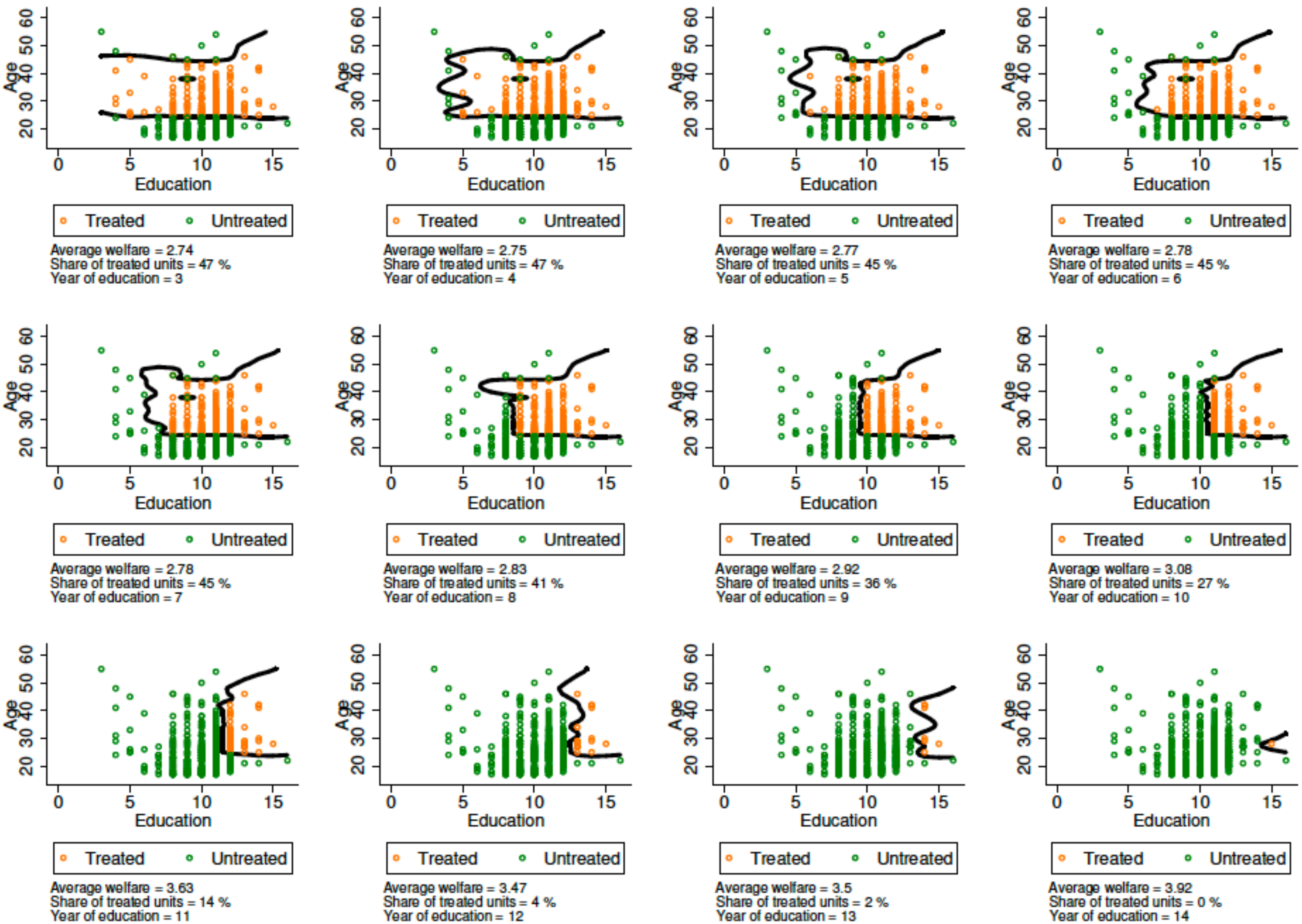
AGE ——————————> **set at its optimal level**

EDUCATION ————> **free to vary**

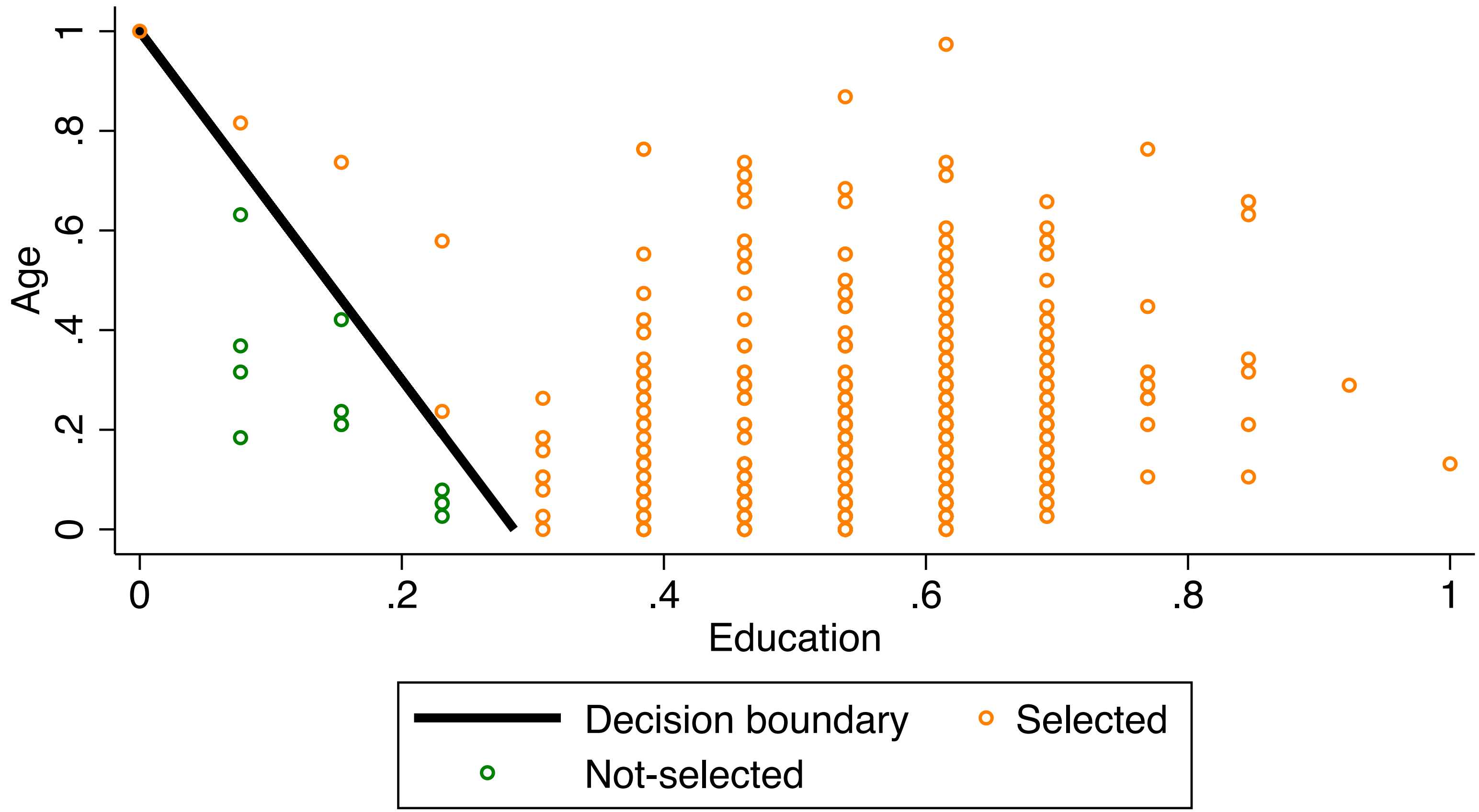
↑

Feature plagued by monotonicity

TRADE-OFFS AND THE "MENU-STRATEGY"



OPTIMAL SELECTION WITH A LINEAR COMBINATION POLICY



Total optimal welfare =748
Total oracle welfare = 764
Regret (absolute) = 15.53
Regret (%) = 2.03
Average welfare = 2.24
Average oracle welfare = 2.23
Share of treated units = 75 %

SOFTWARE

We formed a research group for **OPL software implementation**:

Stata

Cerulli (CNR), `opl` package

R

Guardabascio (Perugia University) and Brogi (Istat)

Python

De Fausti (Istat)

THE STATA PACKAGE "OPL" (CERULLI 2023)

The commands of the Stata package OPL

Optimal policy learning with a threshold-based policy

`opl_tb`

Threshold-based optimal policy learning

`opl_tb_c`

Threshold-based policy learning at specific threshold values

Optimal policy learning with a linear-combination policy

`opl_lc`

Linear-combination optimal policy learning

`opl_lc_c`

Linear-combination policy learning at specific parameters' values

Optimal policy learning with a decision-tree policy

`opl_dt`

Decision-tree optimal policy learning

`Opl_dt_c`

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

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*.

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.

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.

THE “MAKE_CATE” COMMAND

make_cate —

Predicting conditional average treatment effect (CATE) on a new policy based on the training over an old policy

Syntax

make_cate *outcome features* , **treatment(***varname***)** **model(***model_type***)** **new_cate(***name***)** **train_cate(***name***)** **new_data(***name***)**

Description

make_cate is a command generating conditional average treatment effect (CATE) for both a training dataset and a testing (or new) dataset related to a binary (treated vs. untreated) policy program. It provides the main input for runni b opl_tb} (optimal policy learning of a threshold-based policy), **opl_tb_c** (optimal policy learning of a threshold-based policy at specific thresholds), **opl_lc** (optimal policy learning of a linear-combination policy), {helpb opl_lc imal policy learning of a linear-combination policy at specific parameters), **opl_dt** (optimal policy learning of a decision-tree policy), **opl_dt_c** (optimal policy learning of a decision-tree policy at specific thresholds and select ables). Based on Kitagawa and Tetenov (2018), the main econometrics supported by these commands can be found in Cerulli (2022).

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
```

Policy class: Threshold-based

Main results

Learner = Regression adjustment
N. of units = 178
Threshold value c1 = .60000002
Average unconstrained welfare = 2.0673337
Percentage of treated = 1.1
N. of untreated = 176

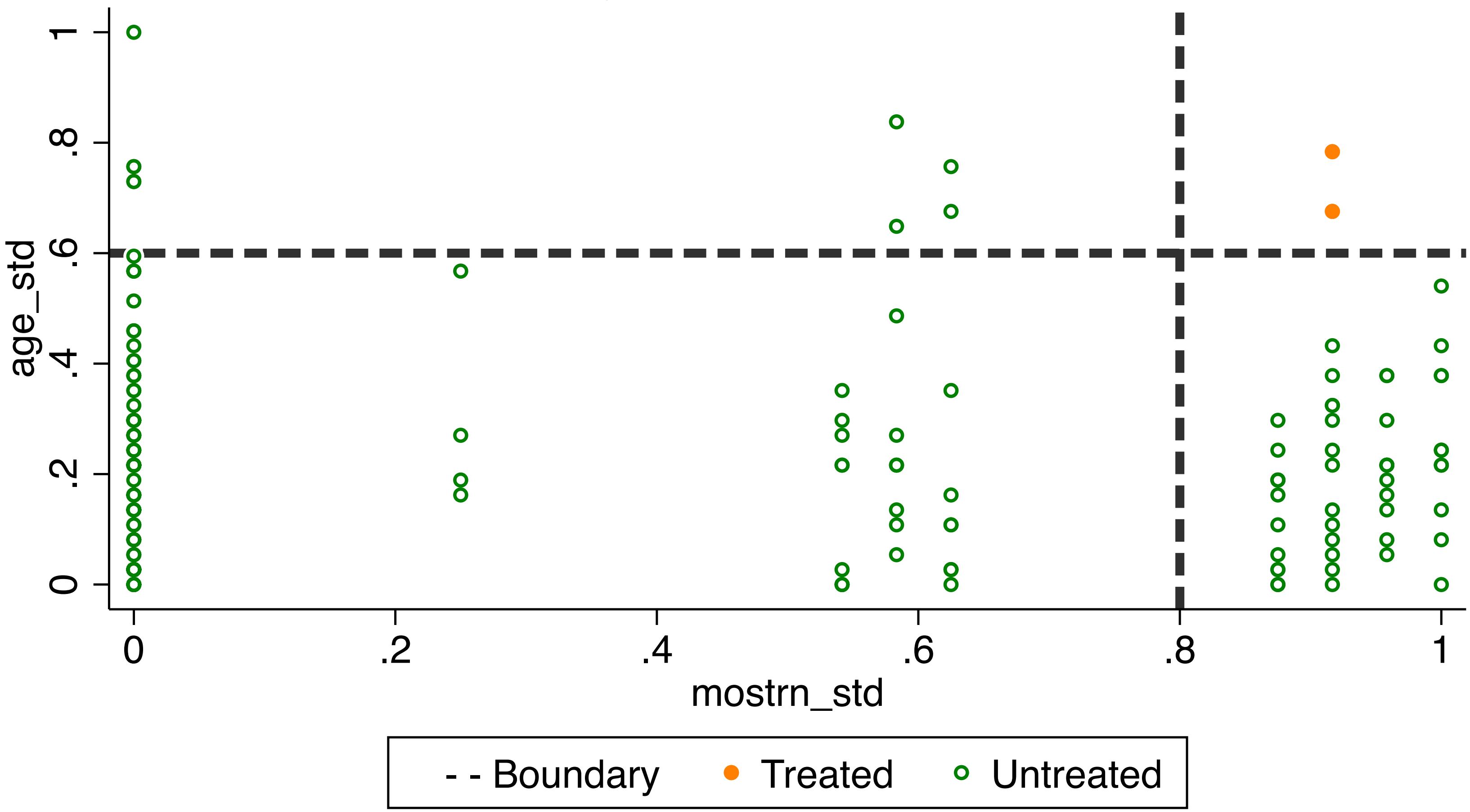
Target variable =
Selection variables = age mostrn
Threshold value c2 = .79999999
Average constrained welfare = 2.885844
N. of treated = 2

. 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**	

Optimal policy assignment

Policy class: threshold-based



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

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
```

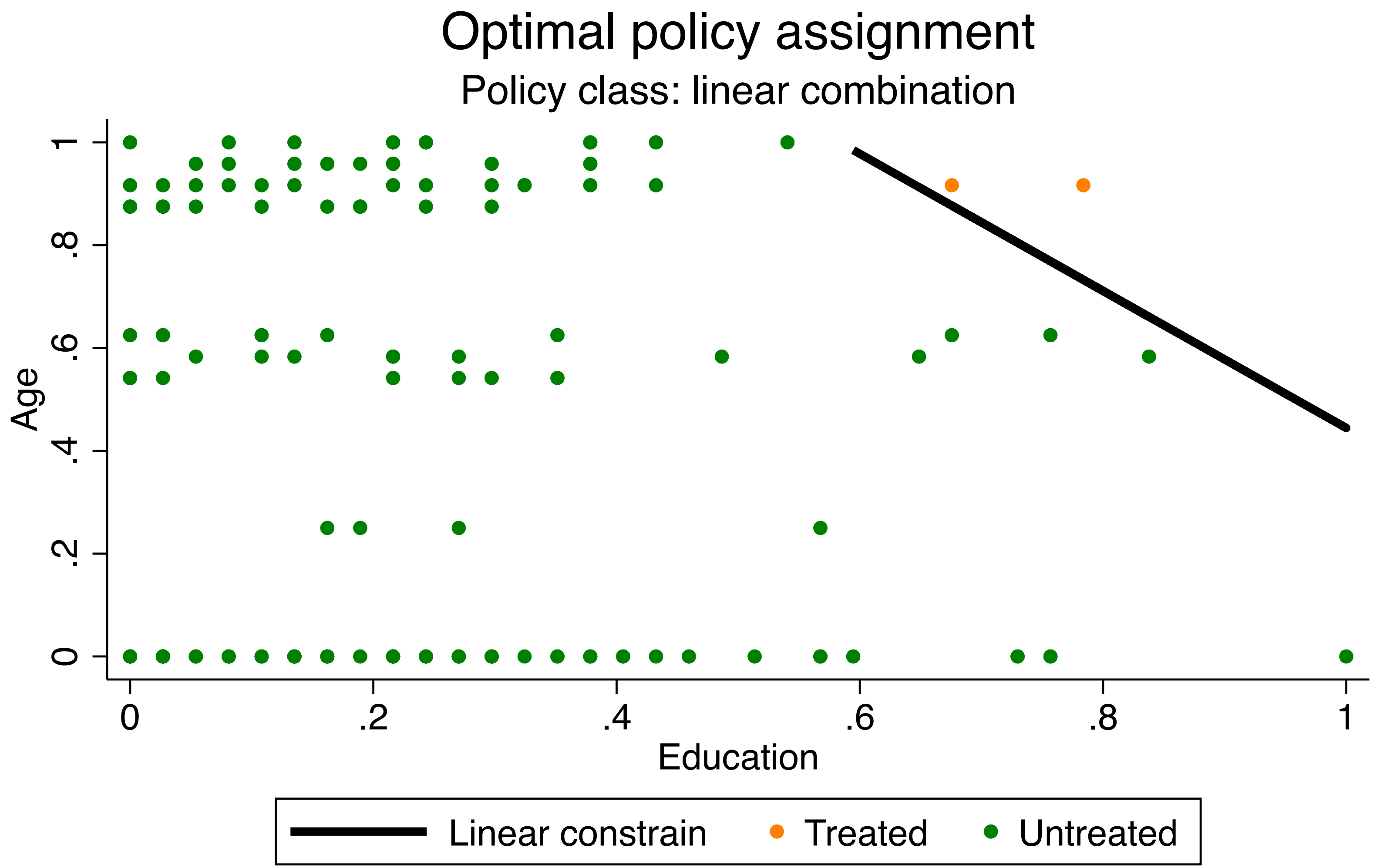
Policy class: Linear-combination

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	



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

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
```

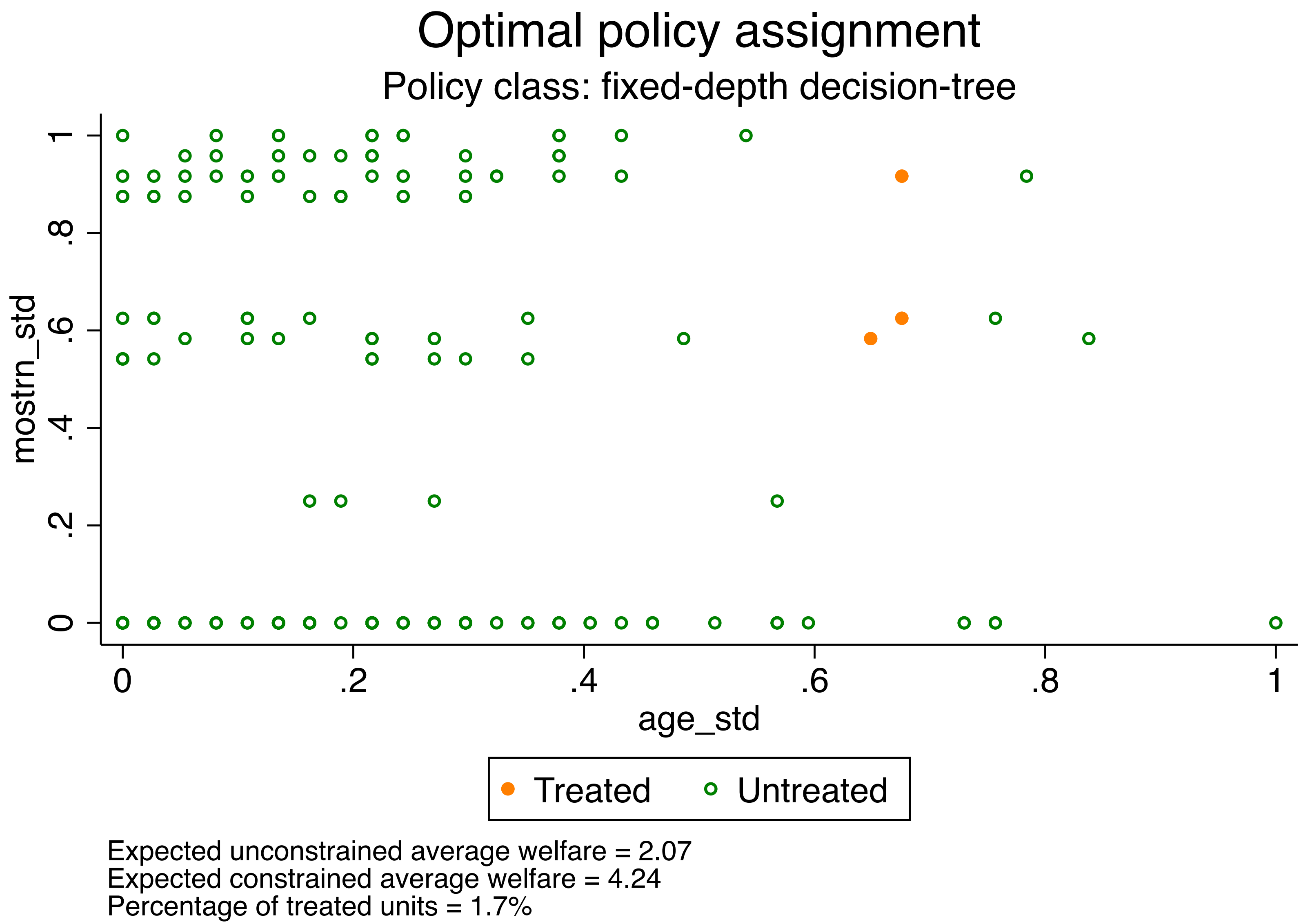
Policy class: Fixed-depth decision-tree

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	


Optimal policy assignment
Policy class: fixed-depth decision-tree
mostrn_std
0
.2
.4
.6
.8
1
age_std
Treated
Untreated
Expected unconstrained average welfare = 2.07
Expected constrained average welfare = 4.24
Percentage of treated units = 1.7%

CONCLUSIONS AND FUTURE AVENUES

- Policy Learning: new frontier of econometrics of prog evaluation
- Theory-driven and data-driven approaches can complement
- Extensions to unobservable selection quite straightforward
- Welfare monotonicity and data sparseness major problems
- Monotonicity solved by “menu strategy”
- Generalization to other policy classes

We provided the Stata OPL package for optimal policy learning

Future developments for OPL:

- **Machine Learning algorithms for estimating $\tau(X)$ by integrating `r_c_ml_stata_cv` (Cerulli, 2022) or `pystacked` (Ahrens, Hansen, Schaffer, 2022)**
- **Coding other policy classes**