unicef | for every child

Implementing Causal Machine Learning in

Wednesday: Flanders & the use of the mcf Python package

August / September / **October** 2024

Michael Lechner
Professor of Econometrics | University of St. Gallen | Switzerland

unicef | for every child

The workshop series

Astana Workshop September 30 to October 4, 2024 (10-13, 14:00-17:30)

- Monday
  - Morning: Identification with experiments & selection on observables
  - Afternoon: Discussion of potential programmes to be evaluated
- Tuesday: Causal Machine Learning (theory) (ends at 16:00)
- **Today**
  - **Empirical examples: Active labour market programmes in Flanders**
  - **The mcf package – how to use it & how to interpret the results**
- Thursday: Doing an empirical study in groups with the data introduced in online workshop 4
- Friday: Discussion of programmes to be evaluated continued (core team only)

unicef | for every child

1 | Introduction

2 | Evaluation of Training in Flanders – Introduction, data, programmes, identification

3 | Evaluation of Training in Flanders – Results, optimal policy analysis, conclusions

4 | The *mcf* Python module: Documentation, major classes & methods

5 | The *mcf* Python module: Effect estimation

6 | The *mcf* Python module: Optimal policy

7 | The *mcf* Python module: Effect estimation & optimal policy

unicef | for every child

ELSEVIER

Contents lists available at ScienceDirect

Labour Economics

journal homepage: www.elsevier.com/locate/labeco

Priority to unemployed immigrants? A causal machine learning evaluation of training in Belgium

Bart Cockx[$,#], Michael Lechner[$$,##], Joost Bollens[$$$,*]

[$] *Department of Economics, Ghent University*
[$$] *Swiss Institute for Empirical Economic Research (SEW), University of St. Gallen*
[$$$] *Vlaamse Dienst voor Arbeidsbemiddeling en Beroepsopleiding (VDAB)*

unicef | for every child

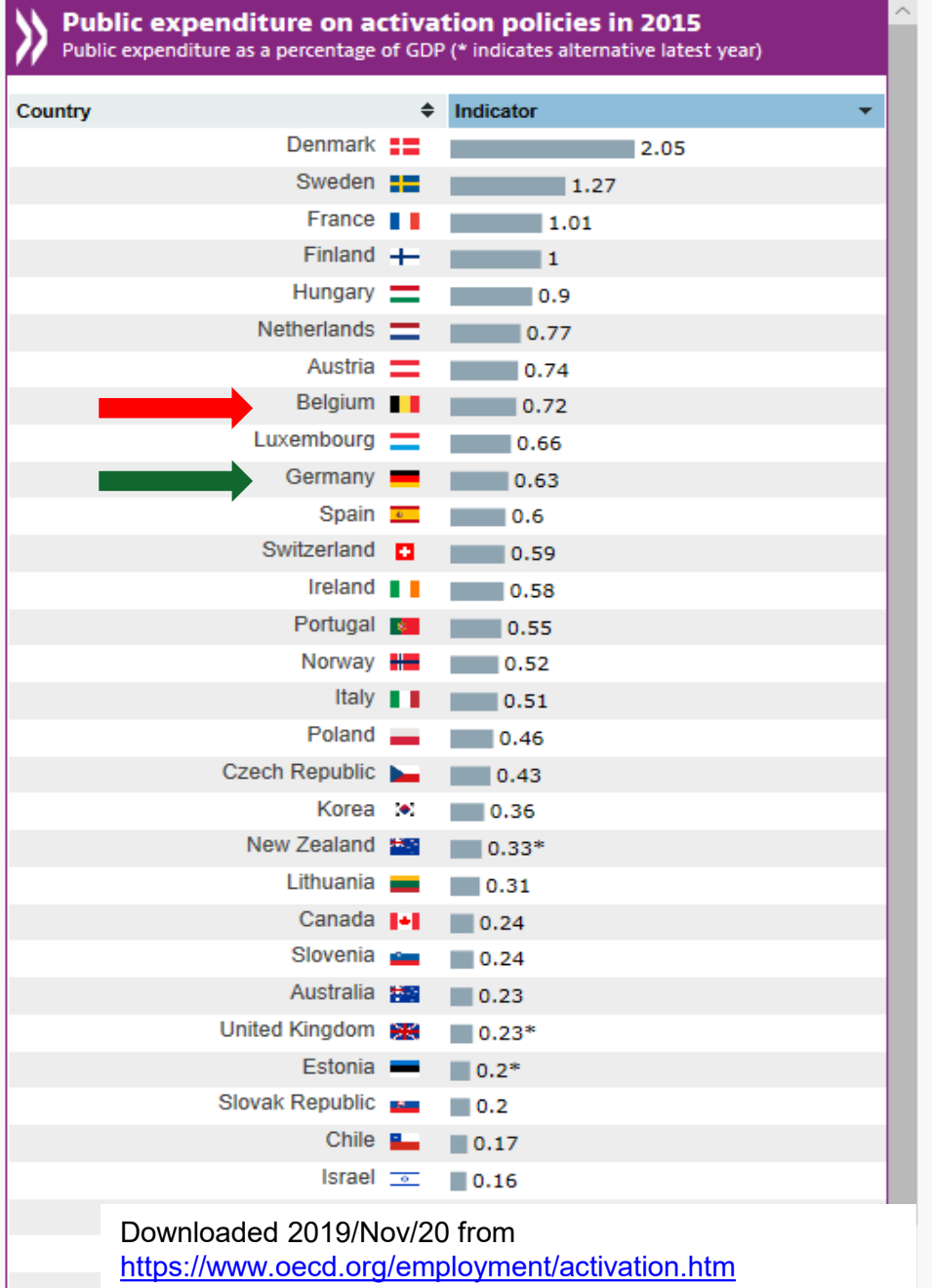
Motivation

Active Labour Market Policies (ALMP) are

Effects of ALMP are heterogenous

- Knowing who benefits by how much has b
  - Improve allocation of UE
  - Better understanding of why some policies h

Causal machine learning methods allow heterogeneities



Downloaded 2019/Nov/20 from
https://www.oecd.org/employment/activation.htm

unicef | for every child

Research questions

Active Labour Market Policies

- What are the effects of important training components of the ALMP in *Flanders*?
- How & by how much do their effects differ for different types of unemployed?
- Can this heterogeneity be used to improve the allocation of unemployed to these programmes?

unicef | for every child

Why Flanders ?

Data access & cooperation with the Public Employment Service (PES) of Flanders

- PES might consider using similar methods to improve their ALMP

External validity

- Inside Flanders: Data consists of very recent Flemish unemployed
- Rest of Europe: ALMP (& PES) of Flanders very similar to ALMPs of many other European countries
- Labour market similar to those of (richer) European countries

unicef | for every child

Literature | Effects of ALMP

This has now become a very large literature in economics

Typically based on **observational studies** informed by rich administrative data employing a **selection-on-observables identification strategy**

Nice summary: Meta study of Card, Kluve, Weber (2018, JEEA)

unicef | for every child

Our approach in this paper

Use very recent, informative administrative data from Flanders such that CIA plausibly holds

Use Modified Causal Forest (MCF) method to investigate the effects of training programmes

- Uncover mean effects & important heterogeneities

Develop simple policy rules that allow to use public resources more efficiently

unicef | for every child

The results of the paper in a nutshell | 1

Mixed findings for average programme effects (before subtracting costs)

- In the long run, all programmes have positive effects
  - But their short-run (lock-in) effects differ

Considerable effect heterogeneity for all programmes

- This heterogeneity appears to go in the same direction for all programmes
  - Non-natives benefit much more, in particular recent immigrants
- Exploiting this heterogeneity for programme allocation suggests possible gains in terms of additional employment
  - Even when using simple, tree-based allocation rules

unicef | for every child

Data | 1

Administrative Data from the Flemish PES

- Records start 1991 & end in 9/2019
- Basic sample: Inflow into UE between Dec 2014 – June 2016
- Final sample size in main analysis
  - No programme participation: 56’324
  - Short (< 6 months) vocational training: 1’305
  - Longer (>= 6 months) vocational training: 1’220
  - Orientation training: 1’115

unicef | for every child

Available information

Labour market status 1991-2019 (monthly/daily)

- Incl. sickness spells & past participation in ALMP

Individual socio-demographics

- Gender, age, education & skills, migration background, position in household, # of kids, …
- Knowledge of Dutch & foreign languages, different drivers licences
- Location of living (district level)
- Sector & job position in last employment prior to relevant UE spell, experience
- Most preferred profession at beginning of current UE spell

Programmes evaluated

Programmes considered in main analysis

- Short (< 6 months) vocational training
- Longer (6 months -10 months) vocational training
- Orientation training

Programme considered in additional analysis

- Intensive counseling
  - Placebo analysis suggested reasonable doubts on credibility of results

Descriptive statistics

Means and standardized differences for selected variables

Variable	No ALMP participation (NOP)	Short vocational training (SVT)		Long vocational training (LVT)		Orientation training (OT)	
Conditioning variables	Sample mean (*standardized difference* 100 relative to NOP*)[1]						
Woman	0.49	0.31	*(36)*	0.40	*(16)*	0.46	*(3)*
Age (in years)	35	34	*(12)*	34	*(12)*	34	*(16)*
Proficiency in Dutch (0-3)[2]	2.4	2.5	*(8)*	2.7	*(40)*	2.6	*(27)*
Months unemployed in the last 10 years	18	19	*(5)*	16	*(11)*	17	*(4)*
Months unemployed in last 2 years	3.9	3.8	*(2)*	3.0	*(18)*	3.2	*(14)*
Education level (1 to 13)	7.2	5.9	*(38)*	7.9	*(22)*	7.2	*(1)*
Availability of labour market history data (LMHD)[3]							
Fraction of people in first unemployment spell (no LMHD)	0.14	0.07	*(23)*	0.05	*(29)*	0.06	*(25)*
Fraction of people with at least 2 years of LMHD	0.80	0.89	*(24)*	0.88	*(24)*	0.88	*(23)*
Fraction of people with at least 10 years of LMHD	0.41	0.44	*(6)*	0.46	*(9)*	0.46	*(9)*
Outcomes							
# of months employed 10 months after starting ALMP[4]	4.0	3.9	*(2)*	2.8	*(33)*	2.4	*(45)*
# of months employed 20 months after starting ALMP[4]	9.8	11	*(16)*	9.4	*(5)*	7.8	*(27)*
# of months employed 30 months after starting ALMP[4]	16	18	*(24)*	17	*(11)*	15	*(12)*
Number of observations	56324	1305		1220		1115	

Notes: [1] The standardized difference is defined as $|\bar{x}^j - \bar{x}^{NOP}|/\sqrt{[Var(x^j) + Var(x^{NOP})]/2} * 100$, where $\bar{x}^j$ and $Var(x^j)$ are the sample mean and variance of the variable x^j for $j \in \{SVT, LVT, OT\}$.

[2] Proficiency in Dutch = 0 if no knowledge; = 1 if limited; = 2 if good; =3 if very good.

[3] This information is implicitly conditioned upon, as it can be obtained by (a combination of) values of conditioning variables (see footnote 6 of Table A.1 in the Online Appendix).

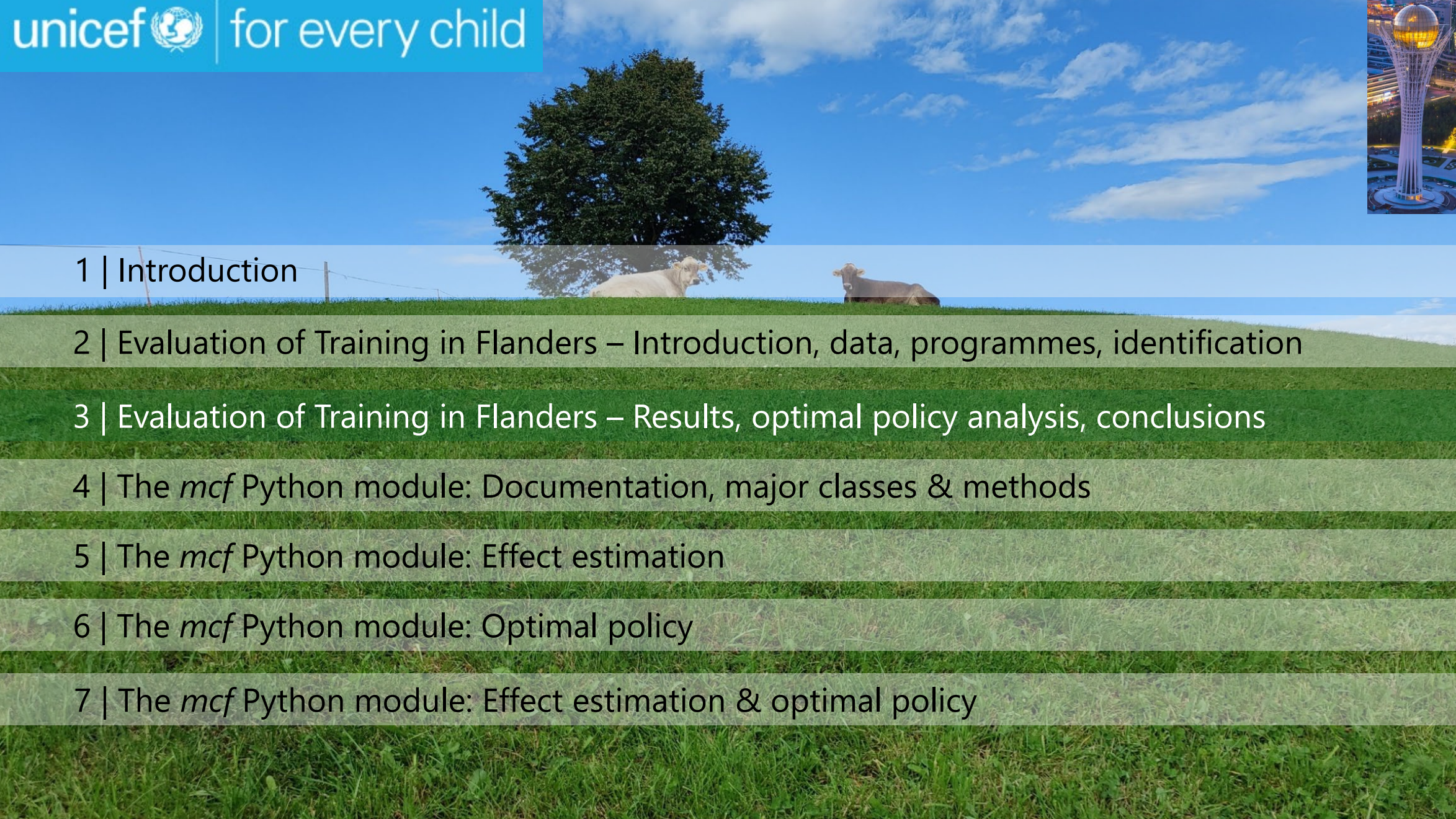
[4] For non-participants in ALMP (NOP) the date at which the ALMP starts (is predicted) (See Section 4.4).

unicef | for every child

Identification

Literature agrees that *unconfoundedness* is usually plausible for evaluation of ALMP with informative, long-term government data

- This is given in this study



1 | Introduction

2 | Evaluation of Training in Flanders – Introduction, data, programmes, identification

3 | Evaluation of Training in Flanders – Results, optimal policy analysis, conclusions

4 | The *mcf* Python module: Documentation, major classes & methods

5 | The *mcf* Python module: Effect estimation

6 | The *mcf* Python module: Optimal policy

7 | The *mcf* Python module: Effect estimation & optimal policy

unicef | for every child

General aspects

Abundance of results (> 300’000 parameters estimated)

- *# of IATE (N x 4 (no of treatments) * 3 / 2)* + # of GATEs + # of ATE
- Concentrate on most important aspects

Main *outcome* variable of interest: Monthly employment

- Either in month *m* or cumulatively over *m* months
- Policy target variable & of economic interest
  - Alternative policy target variable in paper: Unemployment

Comparison to non-participation only

- Other comparisons in paper

ATE & GATE only

- ATET & GATET (partly) in paper

unicef | for every child

Results

Outcome variables

- *Employment* or unemployment in particular month after start of training

Time window: 30 months available for everybody

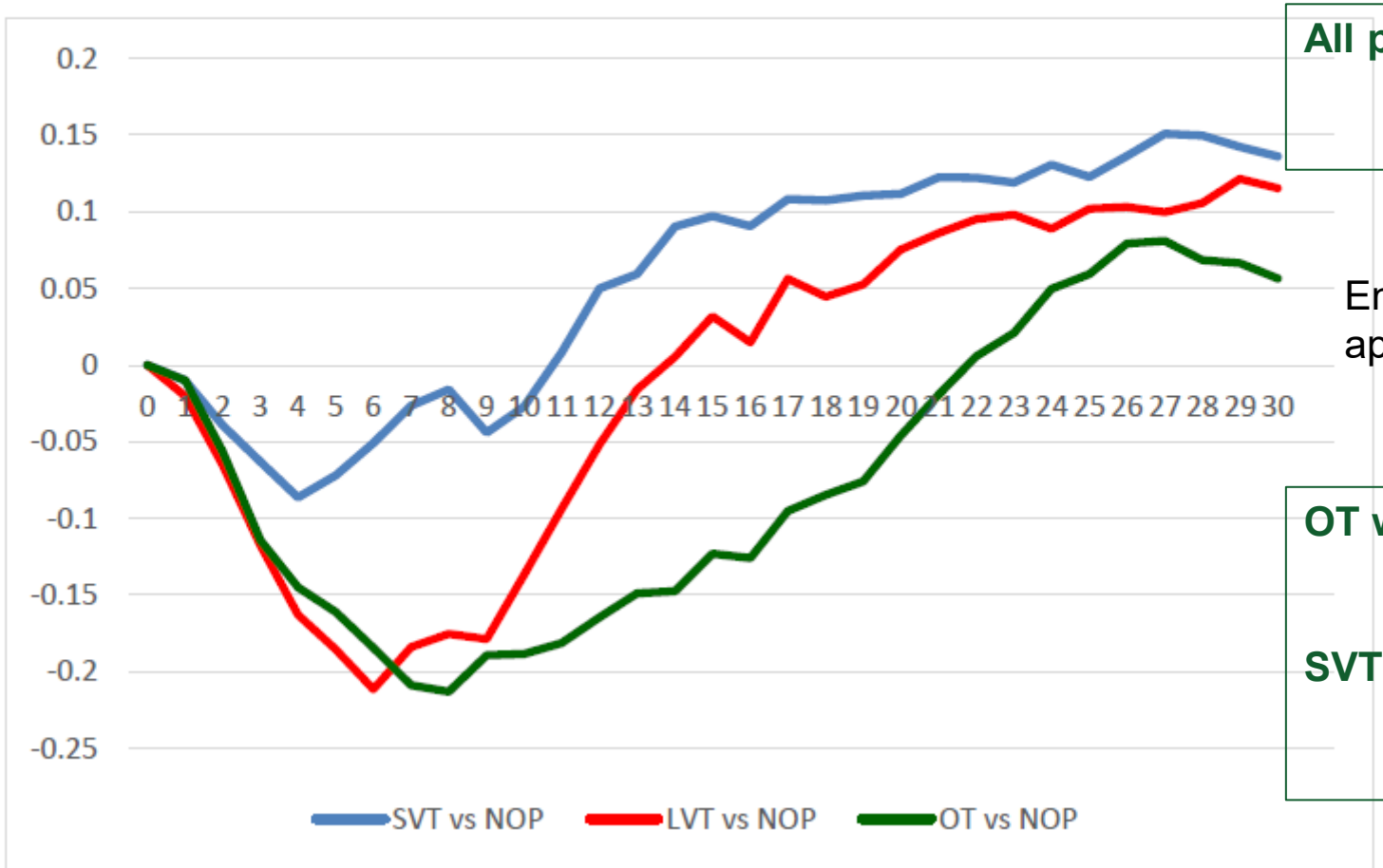
Different aggregation levels over time

- Month by month → dynamics of effect
- Aggregated over certain periods
  - All 30 months: Summary measure for heterogeneity checks
  - Heterogeneity may be different over time
    - Beginning (lock-in period)
    - Last 12 months (effects reached long run level)

Comparisons: Here, comparisons to non-participation only (see paper for more …)

unicef | for every child

Figure 5.1: The time evolution of the ATEs of the employment probability 1 to 30 months after programme start



All programmes have …
- lock-in effects < 0
- long-run effects > 0

Employment level approx. 65%

OT worst
- largest lock-in
- smallest long-run

SVT superior to LVT
- smaller lock-in
- same long-run effect

Other outcome variables

Table 5.1
Effects of the different programmes on cumulative months in employment, unemployment and out of the labour force (ATE)

	No ALMP participation (NOP)	Short vocational training (SVT)	Long vocational training (LVT)	Orientation training (OT)
Cumulative months in *employment 9 months after the programme start*				
NOP	**3.3 (0.02)**			
SVT	-0.1 (0.1)	**3.2 (0.1)**		
LVT	-1.2 (0.1) ***	-1.2 (0.2) ***	**2.1 (0.1)**	
OT	-1.5 (0.1) ***	-1.5 (0.2) ***	-0.3 (0.2)	**1.8 (0.1)**
Cumulative months in *employment between month 22 and month 30* after the programme start				
NOP	**5.7 (0.03)**			
SVT	1.3 (0.1) ***	**7.0 (0.1)**		
LVT	0.9 (0.2) ***	-0.3 (0.2) *	**6.6 (0.1)**	
OT	0.7 (0.2) ***	-0.6 (0.2) ***	-0.3 (0.2)	**6.3 (0.2)**
Cumulative months in *employment 30 months* after the programme start				
NOP	**15.9 (0.1)**			
SVT	2.5 (0.4) ***	**18.4 (0.4)**		
LVT	0.2 (0.4)	-2.3 (0.5) ***	**0.1 (0.4)**	
OT	-1.4 (0.4) ***	-3.9 (0.6) ***	-1.6 (0.6) ***	**14.5 (0.4)**
Cumulative months in *unemployment 30 months* after the programme start				
NOP	**11.2 (0.1)**			
SVT	-1.2 (0.3) ***	**10.0 (0.3)**		
LVT	1.4 (0.4) ***	2.6 (0.5) ***	**12.6 (0.4)**	
OT	2.9 (0.4) ***	4.0 (0.5) ***	1.5 (0.5) ***	**14.1 (0.4)**
Cumulative months out-of-the-labour force 30 months after the programme start				
NOP	**3.2 (0.04)**			
SVT	-1.4 (0.2) ***	**1.7 (0.2)**		
LVT	-1.9 (0.2) ***	-0.5 (0.3) *	**1.2 (0.2)**	
OT	-1.3 (0.2) ***	0.2 (0.3)	0.6 (0.3) **	**1.9 (0.2)**

Note: Outcomes are measured in months. Level of the potential outcome for the specific programme on the main diagonal in bold. All effects are population averages (ATE). Standard errors are in brackets. *, **, *** indicate the precision of the estimate by showing whether the p-value of a two-sided significance test is below 10%, 5%, and 1%, respectively.

unicef | for every child

A priori relevant heterogeneity | GATEs | 1

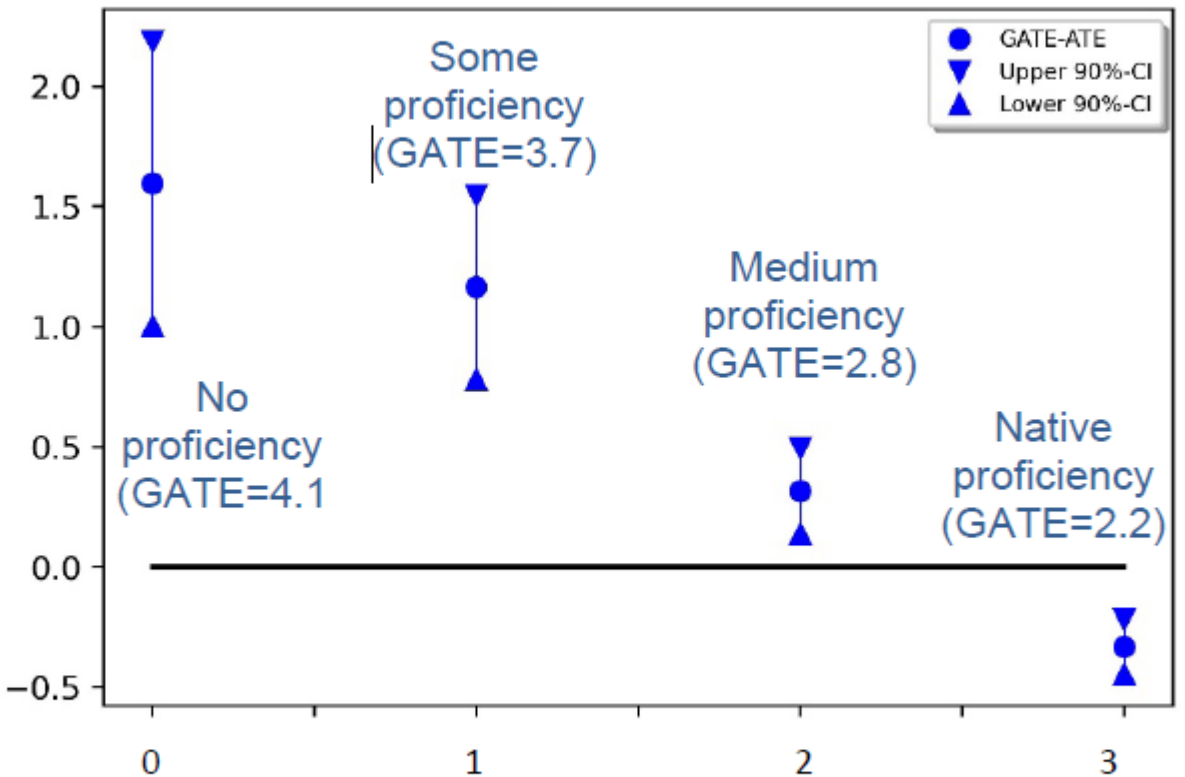
The following variables have been investigated for (univariate) heterogeneity (*p-value of F-Test for effect homogeneity if < 10%*)

- Past unemployment history
- Sex
- Living in city
- Sickness
- Start of programme spell
- Age
- **Dutch Language** (*SVT-NOP: 2%, OT-NOP: 3%*)
- **Country of birth** (*SVT-NOP : 5%, LVT-NOP : 10%*)
- **Education** (*SVT-NOP : 0%*)

Careful interpretation warranted because of multiple-testing

A priori heterogeneity | *GATEs minus ATE* | SVT

Figure 5.2: Difference of GATEs to ATE of SVT relative to NOP for the four proficiency levels in Dutch – Cumulative number of months employed 30 months after programme start



Note: Dutch proficiency displayed on horizontal axis. Vertical axis denotes difference of respective GATE with ATE. (GATE-ATE) and its 90% confidence interval shown. Dutch proficiency varies between no proficiency (0) and native proficiency (3).

A priori heterogeneity | *GATEs minus ATE* | OT

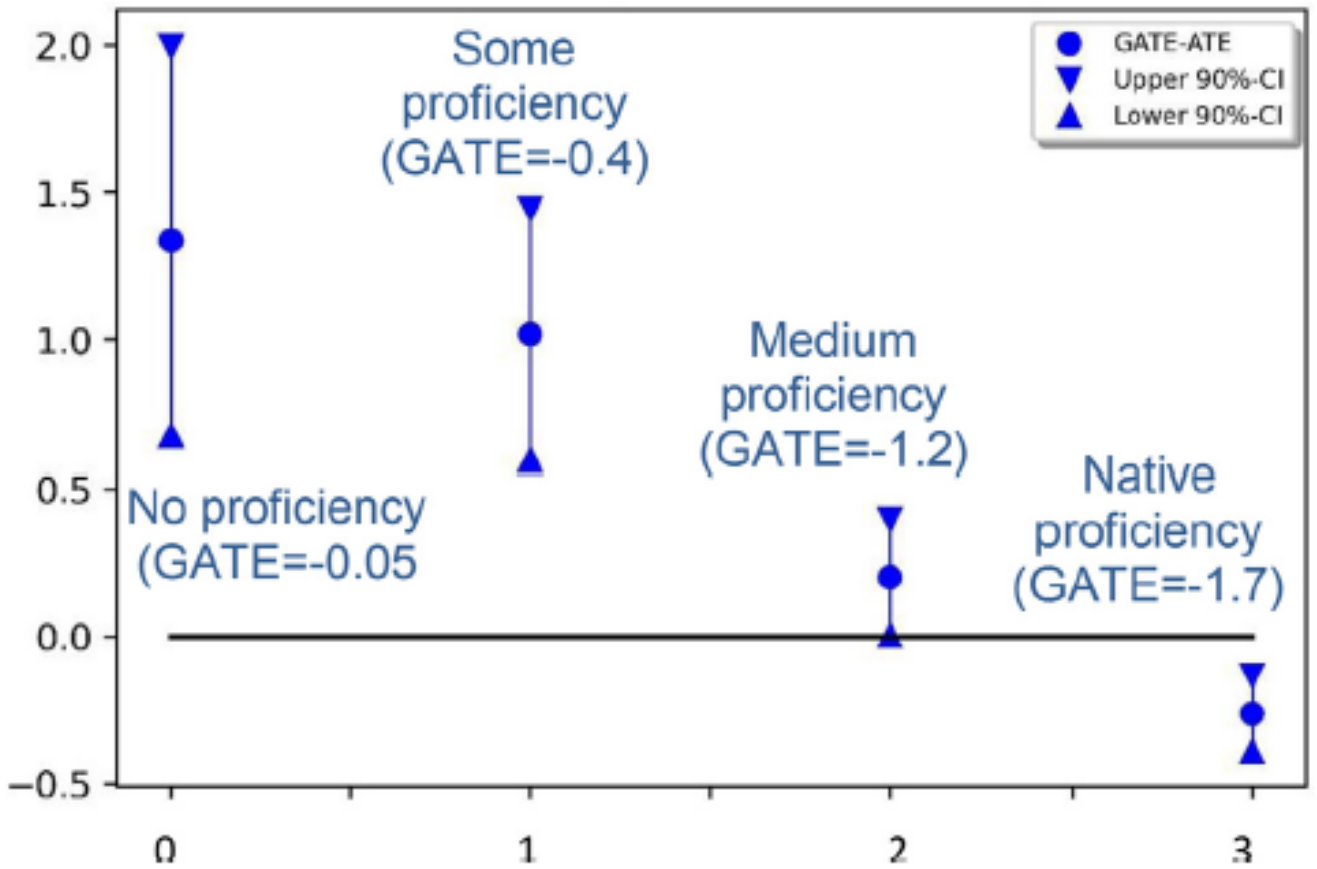
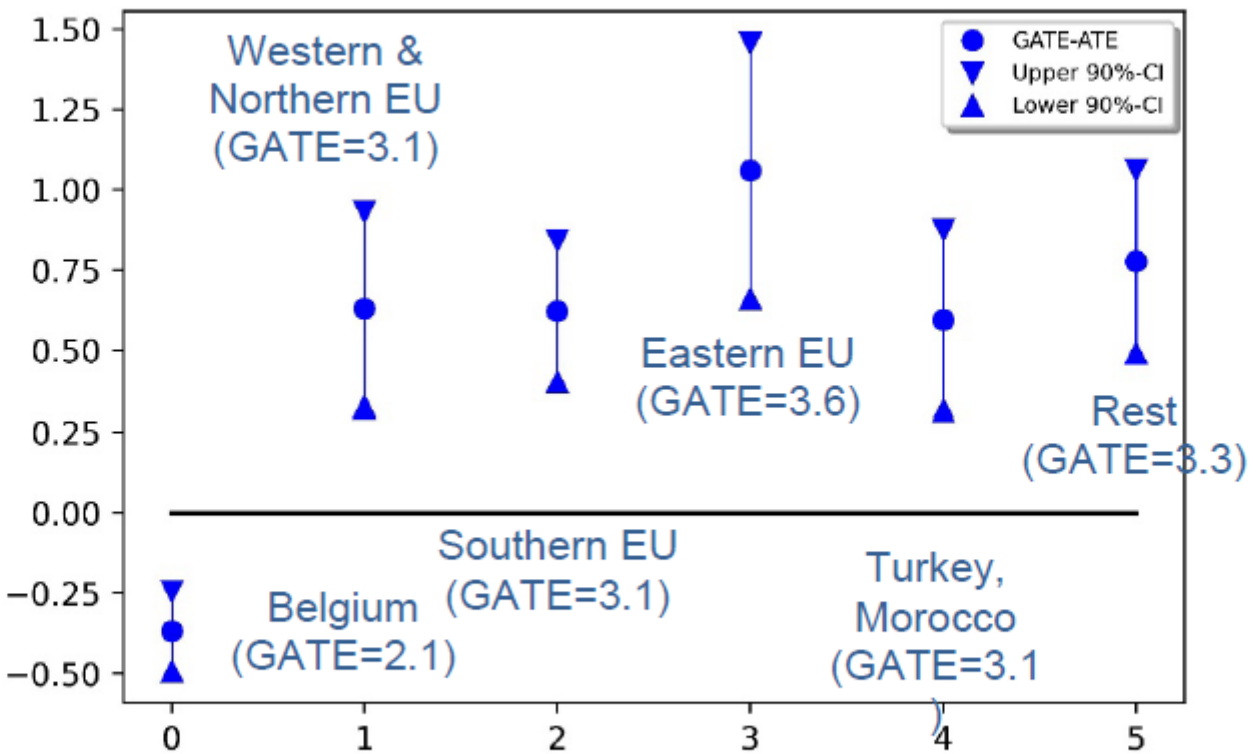


Figure 5.3. Difference of GATEs to ATE of OT relative to NOP for the four proficiency levels in Dutch – Cumulative number of months employed 30 months after programme start

Note: Dutch proficiency is displayed on the horizontal axis. The vertical axis

A priori heterogeneity | *GATEs minus ATE* | SVT

Figure 5.4: Difference of GATEs to ATE of SVT relative to NOP according to country of birth – Cumulative number of months employed 30 months after programme start



Note: Country of birth displayed on horizontal axis. Vertical axis denotes difference of respective GATE with ATE. (GATE-ATE) and its 90% confidence interval shown. The vertical axis measures the deviation of the GATE from the ATE.

unicef | for every child

A priori relevant heterogeneity | **Cumulative Effects**

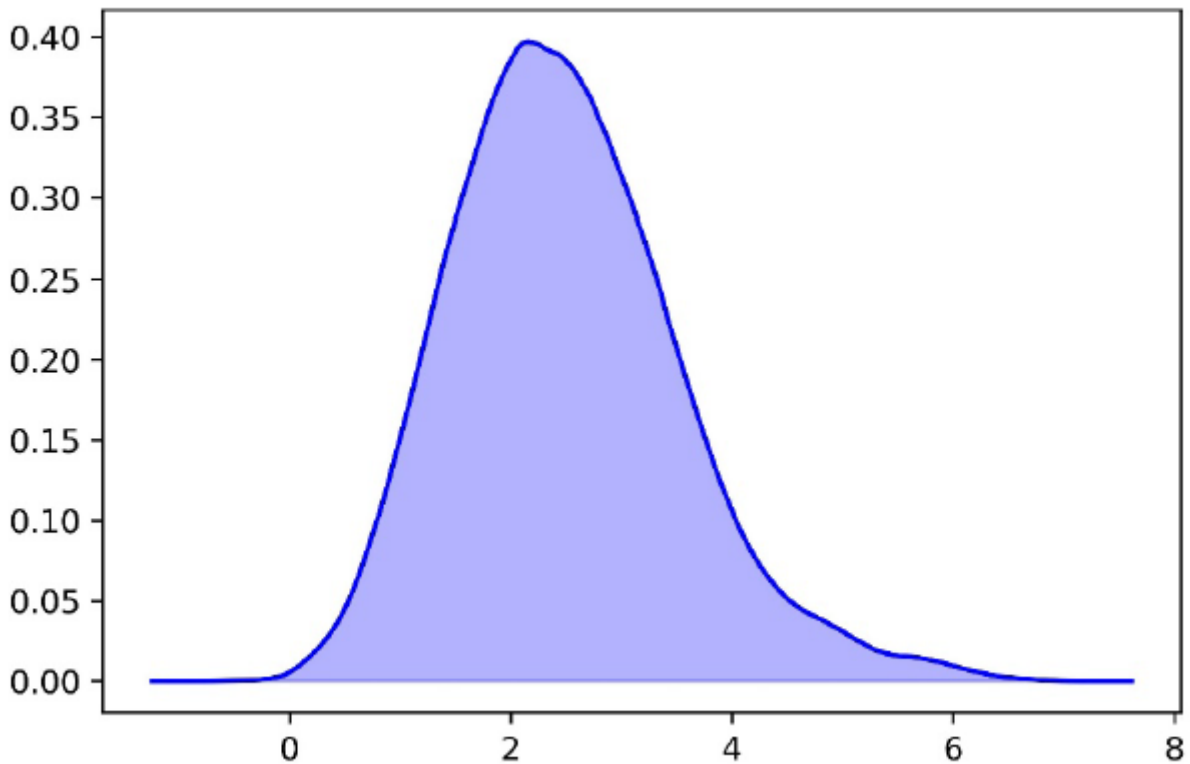
Programmes more effective for UE who are / have …

- ***Lock-in period & long-run***
  - ***Migrants / Bad proficiency in Dutch language***
- Only lock-in period
  - Bad risks in the labour market
    - Older
    - Start programme later
    - Lower education (in particular no vocational training)
    - Worse employment history

No detectible systematic differences for …

- Sickness, sex, living in city

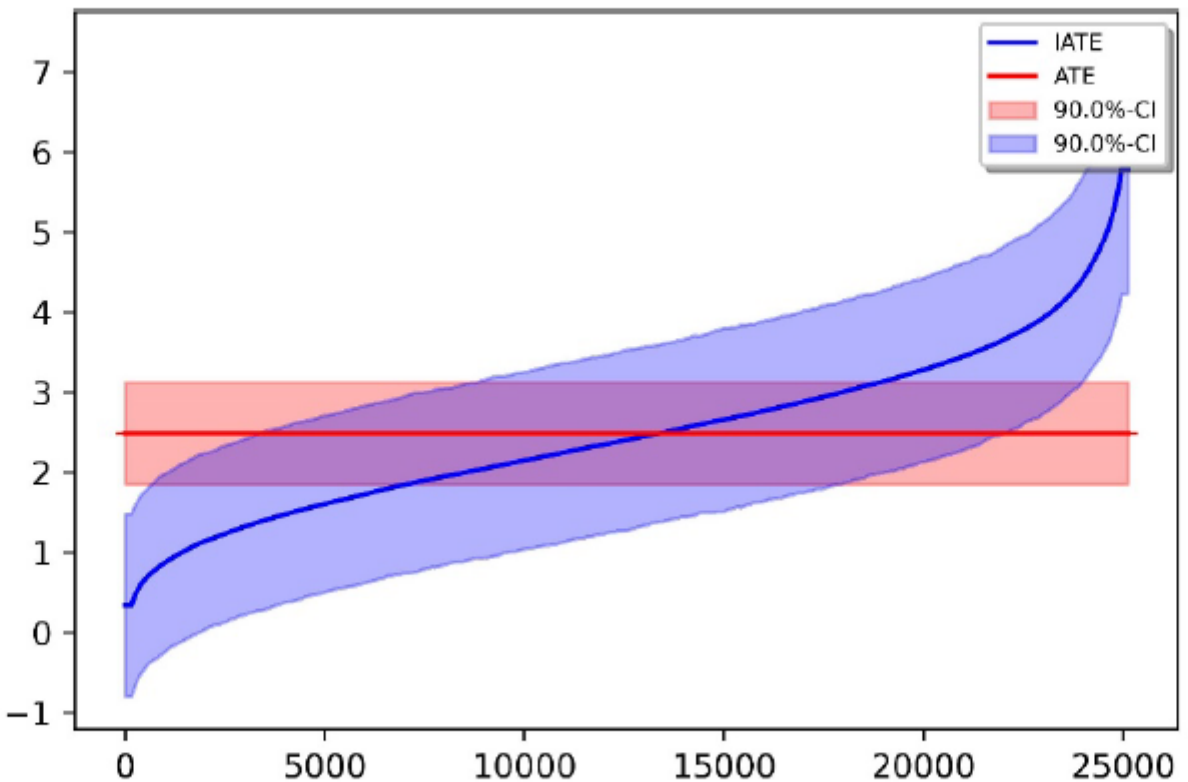
Figure 5.6: Distribution of estimated LATE of SVT vs. NOP



Note: Kernel smooth with Epanechnikov Kernel and Silverman (normality) bandwidth.

Std.: 2.0, *sig. at 5%: 49%*
Std.: 1.4, *sig. at 5%: 2%*
Std.: 1.5, *sig. at 5%: 13%*

Figure 5.7: Overall heterogeneity: sorted effects of SVT relative to NOP – Employment 30 month after the programme start



Note: IATEs are sorted according their size. 90%-confidence interval of IATEs based on estimated standard errors and normal distribution. Standard errors are smoothed by Nadaraya-Watson regression (Epanechnikov kernel with Silverman bandwidth).

unicef | for every child

Detecting heterogeneity | Clustering

IATEs (programme relative to non-participation) are (jointly) clustered into 8 different groups

- K-means++ algorithm
- Clustering based on estimated effects (*IATE(x)*)

Compare mean (& std) of features across groups

Results

- Clustering monotone in effectiveness of all programmes
- Clear differences in features

Table 5.3: Descriptive statistics of clusters based on k-means clustering

Cluster	Least beneficial	2	3	4	5	Most beneficial
	Mean					
	Individualized average treatment effects (IATE) for the comparison to NOP (no participation)					
SVT-NOP	**1.4**	1.7	2.4	2.7	3.4	**4.7**
LVT-NOP	**-0.8**	0.4	-0.2	1.0	0.7	**1.1**
OT-NOP	**-2.2**	-2.1	-1.3	-1.5	-0.5	**0.4**
	Selected features					
Age	**28**	36	33	41	38	**40**
Women (in %)	**33**	52	39	56	51	**55**
Living in a city (in %)	**24**	25	38	34	47	**47**
Proficiency in Dutch (3: high, 0: none)	**2.8**	2.8	2.5	2.7	2.1	**1.5**
Country of birth: Belgium (in %)	**96**	92	66	69	34	**18**
Country of birth: Western & Northern EU (in %)	**1**	2	4	10	12	**15**
Country of birth: Southern EU (in %)	**0**	1	1	1	2	**3**
Country of birth: Eastern EU (in %)	**0**	1	5	3	10	**18**
Country of birth: Turkey & Morocco (in %)	**0**	1	5	5	11	**10**
Country of birth: Rest of the World (in %)	**2**	3	18	12	31	**35**
RIT (unemployed at first labour market entry: in %)	**74**	29	42	15	12	**2**
# of months unemployed in last *10 years* prior to UI	**18**	12	27	15	25	**12**
# of months employed in last *10 years* prior to UI	**45**	80	42	76	41	**18**
# of months unemployed in last *2 years* prior to UI	**5**	2	6	2	4	**2**
# of months employed in last *2 years* prior to UI	**17**	21	15	20	15	**8**
# of days until programme start	**63**	77	92	105	124	**165**
Predicted outcome without programme (NOP)	**16.5**	16.3	15.8	15.7	15.3	**14.9**

Used for clustering

Note: Outcome variable is cumulative employment in the 30 months after programme start. All IATEs for all comparisons to nonparticipation are used to form the 8 clusters. Covariates are not used to form clusters. K-means ++ algorithm used (Arthur and Vassilvitskii, 2007).

Placebo analysis (using the previous UE spell)

Table 7.1
Placebo Effects for the different future programmes on cumulative months in employment, unemployment and out of the labour force (ATE)

	No ALMP participation (NOP)	Short vocational training (SVT)	Long vocational training (LVT)	Orientation training (OT)
Cumulative months in employment 9 months after entry in the preceding unemployment spell				
NOP	**3.9 (0.1)**			
SVT	0.01 (0.3)	**3.9 (0.3)**		
LVT	0.5 (0.3)	0.5 (0.4)	**4.3 (0.3)**	
OT	0.001 (0.4)	-0.02 (0.4)	0.5 (0.4)	**3.9 (0.2)**
Cumulative months in unemployment 9 months after entry in the preceding unemployment spell				
NOP	**4.8 (0.04)**			
SVT	-0.1 (0.3)	**4.8 (0.3)**		
LVT	-0.4 (0.3)	-0.5 (0.4)	**4.4 (0.3)**	
OT	-0.002 (0.3)	-0.1 (0.4)	0.4 (0.4)	**4.8 (0.3)**
Cumulative months out of the labour force 9 months after entry in the preceding unemployment spell				
NOP	**0.4 (0.01)**			
SVT	-0.1 (0.1)	**0.3 (0.02)**		
LVT	-0.1 (0.1)	-0.005 (0.1)	**0.3 (0.1)**	
OT	-0.03 (0.1)	0.04 (0.2)	0.04 (0.2)	**0.3 (0.1)**

unicef | for every child

Policy simulations | 1

Use IATEs to reallocate unemployed to different programmes

No explicit cost data: 3 basic scenarios w.r.t. budget constraints

- Unconstraint
- Observed share of all participants as upper bound
- Observed share of participants in particular programme as upper bound

Target variable: Months of employment & (minus) unemployment (equally weighted; summed up in first 30 months)

Two different ways to obtain alternative allocations

- Black-Box approach
  - Optimize potential outcomes generally
- Policy Trees as restricted policy class
  - Find simple rule that reaps in almost all benefits

Table 6.1
Overall performance gain of simulated hypothetical programme allocations (equal weight for an increase in months in employment and reduction in months in unemployment)

	Share of different programmes in %			Performance gain for all		Performance gain for switchers	
	SVT	LVT	OT	Mean	SE	Mean	SE
Observed	2.3	2.0	1.9	-	-	-	-
				Deviation from the observed state			
Random (restricted)	2.3	2.0	1.9	0.012	0.002	0.10	0.018
Black-box – no constraint	98.7	0.4	0.0	1.73	0.004	1.78	0.004
Black-box – no constraint, only significant	76.2	0.3	0.0	1.57	0.005	2.05	0.007
Black-box – constrained, preference to largest gains*)	2.3	2.0	0.2	0.13	0.002	1.29	0.020
Black-box – constrained, first come, first served**)	2.3	2.0	0.5	0.07	0.002	0.68	0.020
Black-box – constrained, preference to lots of past UE***)	2.3	2.0	0.5	0.08	0.002	0.72	0.019
Policy tree 2 levels, constrained	2.4	0.0	0.0	0.11	-	1.31	-
Policy tree 3 levels, constrained	2.4	2.2	0.0	0.12	-	1.11	-
Policy tree 4 levels, constrained	2.0	2.3	0.0	0.11	-	1.07	-

Notes: *Performance gain* measures the equally weighted sum of the average number of the gained months in employment and the average number of lost months in unemployment relative to the NOP or observed state within 30 months of the programme start. *Switchers* are individuals who change their treatment status under the considered rule; *) In case of excess demand for a programme, priority is given to those individuals for whom the difference in performance between the best and the second-best programme is the largest. **) In case of excess demand, individuals are randomly ordered and then assigned in this order to the best programme in which there is still capacity available until all available programme slots are filled; ***) In case of excess demand, priority is given to the highest number of months in unemployment over the last 10 years before entry in unemployment. SE = standard errors based on bootstrapping.

unicef | for every child

Policy simulations | Results | 1

Results

- Caseworkers allocate UEs into programme without taking IATEs into account
- Maximum gain of 3 months of additional employment in 30 months
  - Costly, because share of SVT increases massively
- Limited total gains if programme shares are kept fixed
  - Programmes are too small to change much in overall employment
  - But big relative improvement: **about 0.4 months additional employment overall by reallocating max 10% of UE (max 5% in programme)**

Such a black-box rule is difficult to 'sell' to UE and CW

Table 6.2
Assignment rules of shallow decision trees

Short training (SVT)	Long training (LVT)	Orientation training (OT)	No programme participation (NOP)
Tree depth = 2			
Worked ≤ 6 months in last 10 years Born in Eastern EU or Turkey or Morocco	Nobody	Nobody	All others
Tree depth = 3			
Worked ≤ 20 months in last 2 years Age less than 40 years Born in Eastern EU, or 'Rest'	Worked ≤ 20 months in last 2 years Age at least 40 years Last employment in specific sectors	Nobody	All others
Tree depth = 4			
Age less than 34 years Did not work in last 2 years Not born in Belgium No knowledge of Dutch	Age at least 34 years Did not work in last 2 years Worked less than 104 months in last 10 years Last employment in specific sectors	Nobody	All others

Note: For the sake of brevity, we do not list the specific sectors selected by the decision tree.

unicef | for every child

Conclusions | 1

Mixed findings for average programme effects

- Short vocational training works
  - Only very small group does not benefit
- Long vocational training is more doubtful overall
  - Lock-in effect large, but also positive long-run effects
  - Considerable underlying effect heterogeneity
- Orientation training
  - On average negative effects because of large lock-in effects
  - However, for recently immigrated foreigners it appears to work

unicef | for every child

Conclusions | 2

Heterogeneity

- Considerable heterogeneity for all programmes
- Appears to have the same direction for all programmes
  - Non-natives benefit more, in particular recent immigrants

Exploiting this heterogeneity for programme allocation suggests that gains exist by allocating unemployed to programmes differently

unicef | for every child

1 | Introduction

2 | Evaluation of Training in Flanders – Introduction, data, programmes, identification

3 | Evaluation of Training in Flanders – Results, optimal policy analysis, conclusions

4 | The *mcf* Python module: Documentation, major classes & methods

5 | The *mcf* Python module: Effect estimation

6 | The *mcf* Python module: Optimal policy

7 | The *mcf* Python module: Effect estimation & optimal policy

unicef | for every child

https://mcfpy.github.io/mcf/#/

The *mcf* homepage

mcf 0.7.1 documentation Getting started User Guide Algorithm Reference Python API FAQ More

Modified Causal Forests

Welcome to the documentation of **mcf**, the Python package implementing the Modified Causal Forest introduced by Lechner (2018). This package allows you to estimate heterogeneous treatment effects for binary and multiple treatments from experimental or observational data. Additionally, mcf offers the capability to learn optimal policy allocations.

If you're new to the **mcf** package, we recommend following these steps:

- Installation Guide: Learn how to install mcf on your system.
- Usage Example: Explore a simple example to quickly understand how to apply mcf to your data.
- Getting started: Dive into a more detailed example to get a better feel for working with mcf.

For those seeking further information:

- The User Guide offers explanations on additional features of the mcf package and provides several example scripts.
- Check out the Python API for details on interacting with the mcf package.
- The Algorithm Reference provides a technical description of the methods used in the package.

This should be done & tested before the course

Installation

Install *Anaconda*

- I am using the full *Anaconda Distribution*

Create an environment inside Anaconda with Python 3.12

conda install sypder [if you use Spyder; or any other conda-based IDE]

pip install mcf

Modified Causal Forests — mcf 0.7.1 documentation (mcfpy.github.io)

Major classes

*ModifiedCausalForest(**keywords)*

- Effect estimation

OptimalPolicy(**keywords)

- Optimal allocation based on predefined policy score

McfOptPolReport(**keywords)

- Creates a pdf-file with results & explanations

`train` (data_df)	Build the modified causal forest on the training data.
`predict` (data_df)	Compute all effects given a causal forest estimated with `train()` method.
`analyse` (results)	Analyse estimated IATE with various descriptive tools.

Major classes & methods of version 0.7.1 | 1

ModifiedCausalForest

- train(training_data_as_DataFrame)
  - Trains the causal forest
  - Requires data on *Y, D, X*
- predict(prediction_data_as_DataFrame)
  - Uses the trained forest to predict the effects with (potentially) new data
  - Requires data on *X*
- analyse(results)
  - Analyses the estimated IATE(x) with various tools
  - Requires 'results' which is the return from the predict method

unicef | for every child

Major classes & methods of version 0.7.1 | *ModifiedCausalForest* | 1

`ModifiedCausalForest.train(data_df)`

Build the modified causal forest on the training data.

Parameters:

data_df (*DataFrame*) – Data used to compute the causal forest. It must contain information about outcomes, treatment, and features.

Returns:

- **tree_df** (*DataFrame*) – Dataset used to build the forest.
- **fill_y_df** (*DataFrame*) – Dataset used to populate the forest with outcomes.
- **outpath** (*String*) – Location of directory in which output is saved.

unicef | for every child

Major classes & methods of version 0.7.1 | *ModifiedCausalForest* | 2

ModifiedCausalForest.predict(*data_df*)

Compute all effects given a causal forest estimated with `train()` method.

Parameters:

data_df (*DataFrame*) – Data used to compute the predictions. It must contain information about features (and treatment if effects for treatment specific subpopulations are desired as well).

Returns:

- **results** (*Dictionary.*) – Results. This dictionary has the following structure: 'ate': ATE, 'ate_se': Standard error of ATE, 'ate effect_list': List of names of estimated effects, 'gate': GATE, 'gate_se': SE of GATE, 'gate_diff': GATE minus ATE, 'gate_diff_se': Standard error of GATE minus ATE, 'cbgate': cbGATE (all covariates balanced), 'cbgate_se': Standard error of CBGATE, 'cbgate_diff': CBGATE minus ATE, 'cbgate_diff_se': Standard error of CBGATE minus ATE, 'bgate': BGATE (only prespecified covariates balanced), 'bgate_se': Standard error of BGATE, 'bgate_diff': BGATE minus ATE, 'bgate_diff_se': Standard errror of BGATE minus ATE, 'gate_names_values': Dictionary: Order of gates parameters and name and values of GATE effects. 'iate': IATE, 'iate_se': Standard error of IATE, 'iate_eff': (More) Efficient IATE (IATE estimated twice and averaged where role of tree_building and tree_filling sample is exchanged), 'iate_data_df': DataFrame with IATEs, 'iate_names_dic': Dictionary containing names of IATEs, 'bala': Effects of balancing tests, 'bala_se': Standard error of effects of balancing tests, 'bala_effect_list': Names of effects of balancing tests
- **outpath** (*String*) – Location of directory in which output is saved.

unicef | for every child

Major classes & methods of version 0.7.1 | *ModifiedCausalForest* | 3

ModifiedCausalForest.analyse(*results*)

Analyse estimated IATE with various descriptive tools.

Parameters:

results (*Dictionary*) – Contains estimation results. This dictionary must have the same structure as the one returned from the `predict()` method.

Raises:

ValueError – Some of the attribute are not compatible with running this method.

Returns:

- **results_plus_cluster** (*Dictionary*) – Same as the results dictionary, but the DataFrame with estimated IATEs contains an additional integer with a group label that comes from k-means clustering.
- **outpath** (*String*) – Location of directory in which output is saved.

unicef | for every child

Major classes & methods of version 0.7.1 | *OptimalPolicy* | 1

solve(training_data_as_DataFrame)

- Train the optimal assignment algorithm

allocate(allocation_data_as_DataFrame)

- Allocate (possible new) data to optimal treatment state

evaluate(allocations_as_df, allocation_data_as_DataFrame)

- Evaluate the allocation

evaluate_multiple(allocations_as_dic, allocation_data_as_DataFrame)

- Evaluate several allocations jointly (nicer output than man single calls to *evaluate*)

`solve`(data_df[, data_title])	Solve for optimal allocation rule.
`allocate`(data_df[, data_title])	Allocate observations to treatment state.
`evaluate`(allocation_df, data_df[, ...])	Evaluate allocation with potential outcome data.
`evaluate_multiple`(allocations_dic, data_df)	Evaluate several allocations simultaneously.

Major classes & methods of version 0.7.1 | *OptimalPolicy* | 2

```
OptimalPolicy.solve(data_df, data_title='')
```

Solve for optimal allocation rule.

Parameters:

- **data_df** (*DataFrame*) – Input data to train particular allocation algorithm.
- **data_title** (*String, optional*) – This string is used as title in outputs. The default is ''.

Returns:

- **allocation_df** (*DataFrame*) – data_df with optimal allocation appended.
- **result_dic** (*Dictionary*) – Contains additional information about trained allocation rule. Only complete when keyword _int_with_output is True.
- **outpath** (*String*) – Location of directory in which output is saved.

Major classes & methods of version 0.7.1 | *OptimalPolicy* | 3

`OptimalPolicy.allocate(data_df, data_title='')`

Allocate observations to treatment state.

Parameters:

- **data_df** (*DataFrame*) – Input data with at least features or policy scores (depending on algorithm).
- **data_title** (*String, optional*) – This string is used as title in outputs. The default is ''.

Returns:

- **allocation_df** (*DataFrame*) – data_df with optimal allocation appended.
- **outpath** (*String*) – Location of directory in which output is saved.

Major classes & methods of version 0.7.1 | *OptimalPolicy* | 4

`OptimalPolicy.evaluate(allocation_df, data_df, data_title='', seed=12434)`

Evaluate allocation with potential outcome data.

Parameters:

- **allocation_df** (*DataFrame*) – Optimal allocation as outputed by the `solve()` and `allocate()` methods.
- **data_df** (*DataFrame*) – Additional information that can be linked to allocation_df.
- **data_title** (*String, optional*) – This string is used as title in outputs. The default is ''.
- **seed** (*Integer, optional*) – Seed for random number generators. The default is 12434.

Returns:

- **results_dic** (*Dictory*) – Collected results of evaluation with self-explanatory keys.
- **outpath** (*String*) – Location of directory in which output is saved.

unicef | for every child

Major classes & methods of version 0.7.1 | *OptimalPolicy* | 5

`OptimalPolicy.evaluate_multiple(allocations_dic, data_df)`

Evaluate several allocations simultaneously.

Parameters:

- **allocations_dic** (*Dictionary.*) – Contains dataframes with specific allocations.
- **data_df** (*DataFrame.*) – Data with the relevant information about potential outcomes which will be used to evaluate the allocations.

Returns:

outpath – Location of directory in which output is saved.

Return type:

String

Major classes & methods of version 0.7.1 | *McfOptPolReport* | 1

report()

- Create a pdf file with results & additional explanations (additionally to standard results)

`McfOptPolReport.report()`

Create a PDF report using instances of the `ModifiedCausalForest` and `OptimalPolicy` classes and saves the file to a user provided location.

Returns:

outpath – Name and Location of file in which pdf output is saved.

Return type:

String

unicef | for every child

1 | Introduction

2 | Evaluation of Training in Flanders – Introduction, data, programmes, identification

3 | Evaluation of Training in Flanders – Results, optimal policy analysis, conclusions

4 | The *mcf* Python module: Documentation, major classes & methods

5 | The *mcf* Python module: Effect estimation

6 | The *mcf* Python module: Optimal policy

7 | The *mcf* Python module: Effect estimation & optimal policy

Effect estimation | *all_parameters_mcf.py*

Structure of example programme

- Import functions & classes
- Generate data (l. 51; example_data())
- Define keywords (or use defaults) & collect them in dictionary (l.659)
- Initialize instance: mymcf (l. 766)
- Use methods with this instance: train, predict, analyse (& sensitivity: experimental)
- Create pdf

Keywords

Effect estimation | *all_parameters_mcf.py* | Output

After running the example programme, there are is a lot of output!

…/out/ contains a pdf file with the key results & some explanations

…/example/output contains more detailed results

- mcf.py.0.6.0_Summary.txt
  - Summary of results (more detailed information than pdf)
- mcf.py.0.6.0.txt
  - Big file, contains (almost) ALL information
- …/ate_iate, …/common_support, …/gate
  - Plots (as pdf & jpeg) & data used to create plots (useful if you want to create 'nicer' plots)

unicef | for every child

1 | Introduction

2 | Evaluation of Training in Flanders – Introduction, data, programmes, identification

3 | Evaluation of Training in Flanders – Results, optimal policy analysis, conclusions

4 | The *mcf* Python module: Documentation, major classes & methods

5 | The *mcf* Python module: Effect estimation

6 | The *mcf* Python module: Optimal policy

7 | The *mcf* Python module: Effect estimation & optimal policy

Optimal policy | *all_parameters_optpolicy.py*

Structure of example programme

- Import functions & classes
- Generate data (l. 56; example_data())
- Define keywords (or use defaults) & collect them in dictionary (l. 346)
- Loop over several different methods to determine optimal allocations
  - Initialize instance: myopt (l. 386)
  - Use methods with this instance: solve, evaluate, allocate
- Compute how much time was needed for the different steps (l. 404)
- Create pdf (l. 407)

unicef | for every child

Optimal policy | *all_parameters_optpolicy.py* | 2

Keywords

Output

- Different methods used for allocation
  - outputOPTBPS
    - Allocating according to best score
  - outputOPTBPS_CLASSIF:
    - Allocation with classifier (trained on allocation based on best score)
  - outputOPTPT
    - Allocation based on Policy Tree
- For all methods, there are 2 files containing the results
  - OptPolicy.0.7.0.txt, OptPolicy.0.7.0_Summary.txt

unicef | for every child

1 | Introduction

2 | Evaluation of Training in Flanders – Introduction, data, programmes, identification

3 | Evaluation of Training in Flanders – Results, optimal policy analysis, conclusions

4 | The *mcf* Python module: Documentation, major classes & methods

5 | The *mcf* Python module: Effect estimation

6 | The *mcf* Python module: Optimal policy

7 | The *mcf* Python module: Effect estimation & optimal policy

Effect estimation & optimal policy | *mcf_optpol_combined.py*

Combines effect estimation with optimal policy

- *IATEs(d,0; x)* estimated by *mcf* are used as policy scores

Uses all 3 classes sequentially

Same output & keywords as for estimation & optimal policy

New: Additional sample splitting (l. 69)

- Sample used to train *mcf* (40%)
- Sample used to train optimal policy algorithm (40%)
- Sample used to evaluate optimal policy algorithm (20%)

unicef | for every child

Tomorrow: Your own empirical evaluation project

Michael Lechner

Swiss Institute for Empirical Economic Research (SEW)
University of St. Gallen | Switzerland