Modified Causal Forest: Estimation, Optimal Policy Estimation

Section 1: General information

Welcome to the mcf estimation and optimal policy package.
This report provides you with a summary of specifications and results. More detailed information can be found in the respective output files. Figures and data (in csv-format, partly to recreate the figures on your own) are provided in the output path as well.

Output information for MCF ESTIMATION
Path for all outputs:
Q:\SEW\Projekte\MLechner\Projekte und Angebote\Unicef\Kasachstan\Workshops\Astana\Wednesday_examples070/example0
Detailed text output:
Q:\SEW\Projekte\MLechner\Projekte und Angebote\Unicef\Kasachstan\Workshops\Astana\Wednesday_examples070/example0/txtFileWithOutput.txt
Summary text output:
Q:\SEW\Projekte\MLechner\Projekte und Angebote\Unicef\Kasachstan\Workshops\Astana\Wednesday_examples070/example0/txtFileWithOutput_Summary.txt

Output information for OPTIMAL POLICY ANALYSIS
Path for all outputs:
Q:\SEW\Projekte\MLechner\Projekte und Angebote\Unicef\Kasachstan\Workshops\Astana\Wednesday_examples070/examplePT
Detailed text output:
Q:\SEW\Projekte\MLechner\Projekte und Angebote\Unicef\Kasachstan\Workshops\Astana\Wednesday_examples070/examplePT/txtFileWithOutput.txt
Summary text output:
Q:\SEW\Projekte\MLechner\Projekte und Angebote\Unicef\Kasachstan\Workshops\Astana\Wednesday_examples070/examplePT/txtFileWithOutput_Summary.txt

BACKGROUND

ESTIMATION OF EFFECTS
The MCF is a comprehensive causal machine learning estimator for the estimation of treatment effects at various levels of granularity, from the average effect at the population level to very fine grained effects at the (almost) individual level. Since effects at the highler levels are obtained from lower level effects, all effects are internally consistent. Recently, the basic package has been appended for new average effects as well as for an optimal policy module. The basis of the MCF estimator is the the causal forest suggested by Wager and Athey (2018). Their estimator has been changed in several dimensions which are described in Lechner (2018). The main changes relate to the objective function as well as to the aggreation of effects. Lechner and Mareckova (2024) provide the asymptotic guarantees for the MCF and compare the MCF, using a large simulation study, to competing approaches like the Generalized Random Forest (GRF, Athey, Tibshirani, Wager, 2019) and Double Machine Learning (DML, Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey, Robins, 2018, Knaus, 2022). In this comparison the MCF faired very well, in particular, but not only,

for heterogeneity estimation. Some operational issues of the MCF are discussed in Bodory, Busshof, Lechner (2022). There are several empirical studies using the MCF, like Cockx, Lechner, Boolens (2023), for example.

References

- Athey, S., J. Tibshirani, S. Wager (2019): Generalized Random Forests, The Annals of Statistics, 47, 1148-1178.
- Athey, S., S. Wager (2019): Estimating Treatment Effects with Causal Forests: An Application, Observational Studies, 5, 21-35.
- Bodory, H., H. Busshoff, M. Lechner (2023): High Resolution Treatment Effects Estimation: Uncovering Effect Heterogeneities with the Modified Causal Forest, Entropy, 24, 1039.
- Chernozhukov, V., D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, J. Robins (2018): Double/debiased machine learning for treatment and structural parameters, Econometrics Journal, 21, C1-C68.
- Cockx, B., M. Lechner, J. Bollens (2023): Priority to unemployed immigrants? A causal machine learning evaluation of training in Belgium, Labour Economics, 80, Article 102306.
- Knaus, M. (2022): Double Machine Learning based Program Evaluation under Unconfoundedness, Econometrics Journal.
- Lechner, M. (2018): Modified Causal Forests for Estimating Heterogeneous Causal Effects, arXiv.
- Lechner, M. (2023): Causal Machine Learning and its Use for Public Policy, Swiss Journal of Economics & Statistics, 159:8.
- Lechner, M., J. Mareckova (2024): Comprehensive Causal Machine Learning, arXiv.
- Wager, S., S. Athey (2018): Estimation and Inference of Heterogeneous Treatment Effects using Random Forests, Journal of the American Statistical Association, 113:523, 1228-1242.

The optimal policy module offers three (basic) algorithms that can be used to exploit fine grained knowledge about effect heterogeneity to obtain decision rules. The current version is implemented for discrete treatments only.
There is also an option for different fairness adjustments.

The BEST_POLICY_SCORE algorithm is based on assigning the treatment that has the highest impact at the unit (e.g., individual) level. If the treatment heterogeneity is known (not estimated), this will lead to the best possible result. This algorithm is computationally not burdensome. However, it will not be easy to understand how the implied rules depends on the features of the unit. Its statistical properties are also not clear (for estimated treatment heterogeneity) and there is a certain danger of overfitting, which could lead to an unsatisfactory out-of-training-sample performance.

The BPS_CLASSIFIER classifier algorithm runs a classifier for each of the allocations obtained by the BEST_POLICY_SCORE algorithm. One advantage of this approach compared to the BEST_POLICY_SCORE algorithm is that prediction of the allocation of (new) observations is fast because it does not require to recompute the policy score (as it is the case with the BEST_POLICY_SCORE algorithm). The specific classifier is selected among four different classifiers from scikit-learn, namely a simple neural network, two classification random forests with minimum leaf size of 2 and 5, and ADDABoost. The selection is a made according to the out-of-sample performance of the Accuracy Score of scikit-learn.

The POLICY TREE algorithm builds optimal shallow decision trees. While these trees are unlikely to lead to gloablly optimal allocations, and are computationally much more expensive, they have the advantage that the decision rule is much easier to understand and that some statistical properties are known, at least for certain versions of such decision trees (e.g., Zhou, Athey, Wager, 2023). The

basic algorithmic implementation follows the recursive algorithm suggested by Zhou, Athey, Wager (2023) with three (more substantial) deviations (=extensions).
Extension 1: Since using One Hot Encoding for categorical variables may lead to rather extreme leaves for such variables with many different values when building (shallow) trees (splitting one value against the rest), a more sophisticated procedure is used that allows to have several values of the categorical variables on both sides of the split.
Extension 2: Constraints are allowed for. They are handled in a sequential manner: First, an approximate treatment-specific cost vector is obtained and used to adjust the policy score accordingly. Second, trees that violate the constraints are removed (to some extent, optional).
Extensions 3: There are a several options implemented to reduce the computational burden, which are discussed below in the section showing the implementation of the policy score.

References
-Zhou, Z., S. Athey, S. Wager (2023): Offline Multi-Action Policy Learning: Generalization and Optimization, Operations Research, INFORMS, 71(1), 148-183.

Modified Causal Forest: Estimation, Optimal Policy Estimation

Section 2: MCF estimation

METHOD
Standard MCF method used. Nearest neighbour matching performed using the Prognostic Score.
Feature selection not is used.
Local centering is used.
Common support is enforced.

VARIABLES
Outcome: outcome
Treatment: treat (with values 0 1 2)
Ordered confounders: x_cont0, x_cont1, x_cont2

EFFECTS ESTIMATED
Average Treatment Effect (ATE), Individualized Average Treatment Effect (IATE), Efficient IATE

Section 2.1: MCF Training

Training uses 6 CPU cores.

Section 2.1.1: Preparation of training data (mcf training)

METHOD
Variables without variation are removed.
Variables that are perfectly correlated with other variables are removed.
Dummy variables with less than 10 observations in the smaller group are removed.
Rows with any missing values for variables needed for training are removed.

RESULTS
No relevant variables were removed.
Sample size of training data: 1600 (no observations removed).

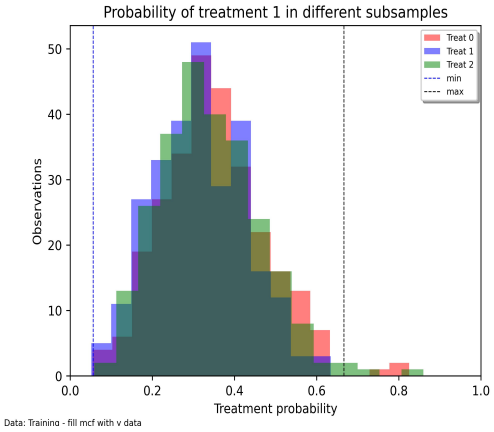
Section 2.1.2: Common support (mcf training)

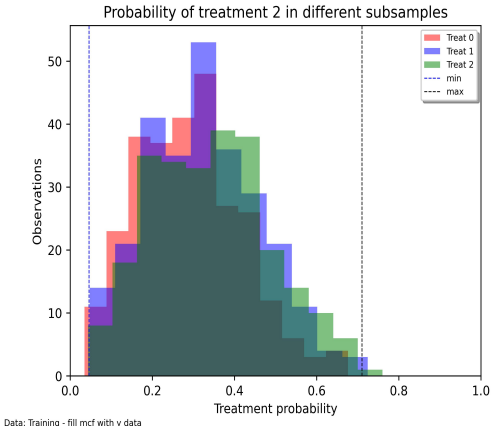
METHOD
The common support analysis is based on checking the overlap in the out-of-sample predictions of the propensity scores (PS) for the different treatment arms. PSs are estimated by random forest classifiers. Overlap is operationalized by computing cut-offs probabilities of the PSs (ignoring the

first treatment arm, because probabilities add to 1 over all treatment arms). These cut-offs are subsequently also applied to the data used for predicting the effects.
Overlap is determined by the min / max rule.
Cut-offs for PS are widened by 0.05.
Out-of-sample predictions are generated by 5-fold cross-validation.

RESULTS
Share of observations deleted: 1.00%
Number of observations remaining: 1584

Common support plots





Section 2.1.3: Local centering (mcf training)

METHOD
Local centering is based on training a regression to predict the outcome variable conditional on the features (without the treatment). The regression method is selected among various versions of Random Forests, Support Vector Machines, Boosting methods, and Neural Networks of scikit-learn. The best method is selected by minimizing their out-of-sample Mean Squared Error using 5-fold cross-validation. The full set of results of the method selection step are contained in Q:\SEW\Projekte\MLechner\Projekte und Angebote\Unicef\Kasachstan\Workshops\Astana\Wednesday_examples070/example0/txtFileWithOutput.txt.
The respective out-of-sample predictions are subtracted from the observed outcome in the training data used to build the forest. These out-of-sample predictions are generated by 5-fold cross-validation.

RESULTS
Out-of-sample fit for Random Forest of Ey|x (R2) for outcome: 5.87%

Modified Causal Forest: Estimation, Optimal Policy Estimation

Section 2.1.4: Forest

METHOD and tuning parameters
Method used for forest building is MSE & MCE Penalty mse_d. MSE is only computed for IATEs comparing all treatments to the first (control) treatment.
The causal forest consists of 1000 trees.
The minimum leaf size is 5.
The number of variables considered for each split is 2
The share of data used in the subsamples for forest building is 67%.
The share of the data used in the subsamples for forest evaluation (outcomes) is 100%.
Alpha regularity is set to 10%.

outcome_lc is the outcome variable used for splitting (locally centered).
The features used for splitting are x_cont0 x_cont1 x_cont2.

RESULTS
Each tree has on average 76.65 leaves.
Each leaf contains on average 6.9 observations. The median # of observations per leaf is 6.
The smallest leaves have 5 observations.
The largest leaf has 35 observations.
17.96% of the leaves were merged when populating the forest with outcomes from the honesty sample.

NOTE
For the estimation of the "efficient" IATEs, the role of the samples used for building the forest and populating it are reversed. Subsequently, the two sets of estimates for the IATEs are averaged.

Modified Causal Forest: Estimation, Optimal Policy Estimation

Section 2: MCF estimation

Section 2.2: MCF Prediction of Effects

Training uses 6 CPU cores.

Section 2.2.1: Common support (mcf prediction)

Share of observations deleted: 0.31%
Number of observations remaining: 1595

Section 2.2.2: Results

GENERAL REMARKS
The following results for the different parameters are all based on the same causal forests (CF). The combination of the CF with the potentially new data provided leads to a weight matrix. This matrix may be large requiring some computational optimisations, such as processing it in batches and saving it in sparse matrix format. One advantage of this approach is that aggregated effects (ATE, GATE) can be computed by aggregation of the weights used for the IATE. Thus a high internal consistency is preserved in the sense that IATE will aggregate to GATEs, which in turn will aggregate to ATEs.

ESTIMATION
Weights of individual training observations are truncated at 5.00%.Aggregation of IATE to ATEs and GATEs may not be exact due to weight truncation.

INFERENCE
Inference is based on using the weight matrix. Nonparametric regressions are based on k-nearest neighbours.

NOTE
Treatment effects for specific treatment groups (so-called treatment effects on the treated or non-treated) can only be provided if the data provided for prediction contains a treatment variable (which is not required for the other effects).

Section 2.2.2.1: ATE

Modified Causal Forest: Estimation, Optimal Policy Estimation

RESULT

ATE for outcome

Comparison	*Effect*	*SE*	*t-value*	*p-value (%)*	*Sig.*
1 vs 0	1.21	0.142	8.52	0.0	****
2 vs 0	1.031	0.146	7.06	0.0	****
2 vs 1	-0.179	0.148	1.21	22.63	

Note: *, **, ***, **** denote significance at the 10%, 5%, 1%, 0.1% level. The results for the potential outcomes can be found in the output files.

Section 2.2.2.2: IATE

This section contains parts of the descriptive analysis of the IATEs. Use the analyse method to obtain more descriptives of the IATEs, like their distribution, and their relations to the features.

METHODOLOGICAL NOTE
In order to increase the efficiency of the IATE estimation, a second set of IATEs is computed by reversing the role of the two samples used to build the forest and to populate it with the outcome information. The two IATEs are averaged to obtain a more precise estimator (which may be particulary useful when the IATEs, or the corresponding potential outcomes, are used as inputs for decision models).
The following descriprive analysis is based on the first round IATEs only.

RESULTS

Outcome variable: outcome

Comparison	Mean	Median	Std	Effect > 0
1 vs 0	1.20470	1.04780	1.01636	86.02%
2 vs 0	1.04867	0.78704	1.17305	78.81%
2 vs 1	-0.15603	-0.19917	0.43857	35.61%

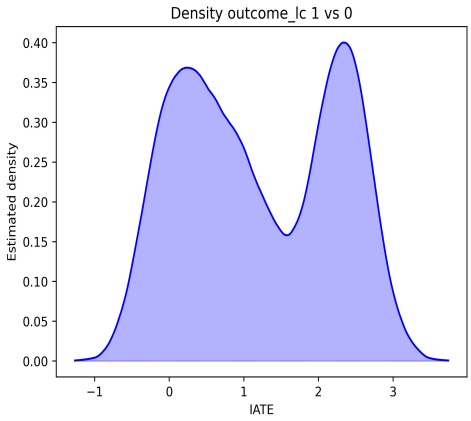
Modified Causal Forest: Estimation, Optimal Policy Estimation

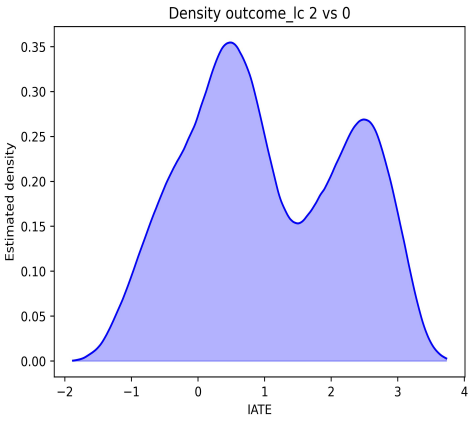
Section 2: MCF estimation

Section 2.3: Analysis of Estimated IATEs

This section contains parts of the descriptive analysis of the IATEs. More detailed tables and figures are contained in the output files and directories. These additional results include variable importance plots from a regression random forest as well as and linear regression results. Both estimators use the estimated IATEs as dependent variables and the confounders and heterogeneity variables as features.

IATEs





K-MEANS CLUSTERING
The sample is divided using k-means clustering based on the estimated IATEs. The number of clusters is determined by maximizing the Average Silhouette Score on a given grid. The following table shows the means of the IATEs, potential outcomes, and the features in these clusters, respectively.

Number of observations in the clusters

Cluster 0: 445
Cluster 1: 550
Cluster 2: 600

Modified Causal Forest: Estimation, Optimal Policy Estimation

IATE

Comparison	*0*	*1*	*2*
outcome_lc1vs0_iate	0.03	0.9	2.36
outcome_lc2vs0_iate	-0.31	0.71	2.37

Note: Mean of variable in cluster.

Potential Outcomes

Comparison	*0*	*1*	*2*
outcome_lc0_un_lc_pot	-0.05	-0.18	-0.2
outcome_lc1_un_lc_pot	-0.02	0.72	2.16
outcome_lc2_un_lc_pot	-0.36	0.53	2.17

Note: Mean of variable in cluster.

Features

Comparison	*0*	*1*	*2*
x_cont0	-0.37	-0.4	0.59
x_cont1	-1.06	0.08	0.7
x_cont2	-0.06	0.03	0.02

Note: Mean of variable in cluster. Categorical variables are recoded to indicator (dummy) variables.

Modified Causal Forest: Estimation, Optimal Policy Estimation

Section 3: Optimal Policy

METHOD
The assignment rule is based on allocating units using a shallow decision tree of depth 4 (based on 2 optimal trees, depth of 1st tree: 2, depth of 2nd tree: 2).

VARIABLES provided
Policy scores: outcome_lc0_un_lc_pot_eff, outcome_lc1_un_lc_pot_eff, outcome_lc2_un_lc_pot_eff
Treatment: treat
Identifier: id_mcfx
Oderered features of units: x_cont0, x_cont1, x_cont2

COSTS
No user provided costs of specific treatments.

RESTRICTIONS of treatment shares
Treatment shares are unrestricted.

FAIRNESS
No fairness adjustments performed.

Section 3.1: Optimal Policy: Training

COMPUTATION
6 logical cores are used for processing.
Continuous variables are internally split for best use of cpu ressources.

DATA PREPARATION
Variables without variation are removed.
Variables that are perfectly correlated with other variables are removed.
Dummy variables with less than 10 observations in the smaller group are removed.
Rows with any missing values for variables needed for training are removed.

COMPUTATIONAL EFFICIENCY
Optimal policy trees are computationally very demanding. Therefore, several approximation parameters are used.
Instead of evaluating all values of continuous variables and combinations of values of categorical variables when splitting, only 100 values are considered. These values are equally spaced for continuous variables and random combinations for categorical variables. This number is used for EVERY splitting decision, i.e. the approximation improves the smaller the data in the leaf becomes. Increasing this value can significantly improve the computational performance at the price of a certain approximation loss.
The depth of the tree is also a key parameter. Usually, it is very hard to estimate trees beyond the depth of 4 (16 leaves) with reasonably sized training data. There are two options to improve the computational performance. The first one is to reduce the depth (leading to loss of efficiency but a

gain in interpretability). The second option is to split the tree building into several steps.
In this application, this two-step tree buildung option is implemented in the following way: After buildung the first tree of depth 2, in each leaf of this tree, a second optimal tree of depth 2 is built. Subsequently, these trees are combined to form the final tree of depth 4. For given final tree depth, the more similar the depths of the two trees are, the faster the algorithm. However, the final tree will of course be subject to an additional approximation error.
Another parameter crucial for performance is the minimum leaf size. Too small leaves may be undesirable for practical purposes (and they increase computation times). The minimum leaf size in this application is set to 20.
In addition, the user may reduce the size of the training data to increase speed, but this will increase sampling noise.

CATEGORICAL VARIABLES
There are two different approximation methods for larger categorical variables. Since we build optimal trees, for categorical variables we need to check all possible combinations of the different values that lead to binary splits. This number could indeed be huge. Therefore, we compare only 200 different combinations. The available methods differ on how these methods are implemented. In this application, at each possible split, we sort the values of the categorical variables according to the values of the policy scores as one would do for a standard random forest. If this set is still too large, a random sample of the entailed combinations is drawn.

STRUCTURE OF FINAL TREE (using data from Training PT data)

Leaf information for estimated policy tree

Depth of 1st tree: 2, depth of 2nd tree: 2, total depth: 4
- -
Leaf 00: x_cont1 <= -0.937 x_cont0 <= 0.917 x_cont0 <= -0.090 x_cont0 <= -0.429
Alloc Treatment: 0 Obs: 152
- -
Leaf 01: x_cont1 <= -0.937 x_cont0 <= 0.917 x_cont0 <= -0.090 x_cont0 > -0.429
Alloc Treatment: 1 Obs: 40
- -
Leaf 10: x_cont1 <= -0.937 x_cont0 <= 0.917 x_cont0 > -0.090 x_cont1 <= -1.523
Alloc Treatment: 2 Obs: 25
- -
Leaf 11: x_cont1 <= -0.937 x_cont0 <= 0.917 x_cont0 > -0.090 x_cont1 > -1.523
Alloc Treatment: 0 Obs: 73
- -
Leaf 20: x_cont1 <= -0.937 x_cont0 > 0.917 x_cont0 <= 1.325 x_cont0 <= 1.152
Alloc Treatment: 2 Obs: 22
- -
Leaf 21: x_cont1 <= -0.937 x_cont0 > 0.917 x_cont0 <= 1.325 x_cont0 > 1.152
Alloc Treatment: 2 Obs: 20
- -
Leaf 30: x_cont1 <= -0.937 x_cont0 > 0.917 x_cont0 > 1.325 x_cont0 <= 1.530
Alloc Treatment: 2 Obs: 22

- -
Leaf 31: x_cont1 <= -0.937 x_cont0 > 0.917 x_cont0 > 1.325 x_cont0 > 1.530
Alloc Treatment: 2 Obs: 21
- -
Leaf 40: x_cont1 > -0.937 x_cont1 <= -0.252 x_cont1 <= -0.607 x_cont0 <= 0.550
Alloc Treatment: 1 Obs: 111
- -
Leaf 41: x_cont1 > -0.937 x_cont1 <= -0.252 x_cont1 <= -0.607 x_cont0 > 0.550
Alloc Treatment: 2 Obs: 48
- -
Leaf 50: x_cont1 > -0.937 x_cont1 <= -0.252 x_cont1 > -0.607 x_cont2 <= 0.743
Alloc Treatment: 1 Obs: 126
- -
Leaf 51: x_cont1 > -0.937 x_cont1 <= -0.252 x_cont1 > -0.607 x_cont2 > 0.743
Alloc Treatment: 2 Obs: 50
- -
Leaf 60: x_cont1 > -0.937 x_cont1 > -0.252 x_cont2 <= -0.888 x_cont0 <= 0.629
Alloc Treatment: 2 Obs: 153
- -
Leaf 61: x_cont1 > -0.937 x_cont1 > -0.252 x_cont2 <= -0.888 x_cont0 > 0.629
Alloc Treatment: 1 Obs: 67
- -
Leaf 70: x_cont1 > -0.937 x_cont1 > -0.252 x_cont2 > -0.888 x_cont2 <= -0.823
Alloc Treatment: 1 Obs: 21
- -
Leaf 71: x_cont1 > -0.937 x_cont1 > -0.252 x_cont2 > -0.888 x_cont2 > -0.823
Alloc Treatment: 2 Obs: 644
- -
NOTE: Splitpoints displayed for ordered variables are midpoints between observable values (e.g., 0.5 for a variable with values of 0 and 1).

Section 3.2: Optimal Policy: Evaluation of Allocation(s)

Main evaluation results.

Note: The output files contain relevant additional information, like a descriptive analysis of the treatment groups.

Evaluation of treatment allocation

Allocation	*Value function*	*Share of 0 in %*	*Share of 1 in %*	*Share of 2 in %*
All Policy Tree	1.0498	14.11	22.88	63.01
All observed	0.5271	33.54	33.54	32.92
All random	0.5974	31.79	34.86	33.35
Switchers Policy Tree	1.0999	13.92	22.51	63.56

Modified Causal Forest: Estimation, Optimal Policy Estimation

Switchers random	0.6573	31.72	34.28	34.0

Note: Allocation analysed is the SAME as the one obtained from the training data.

Evaluation of treatment allocation

Allocation	*Value function*	*Share of 0 in %*	*Share of 1 in %*	*Share of 2 in %*
All Policy Tree	1.045	13.82	23.37	62.81
All observed	0.5718	31.16	33.54	35.3
All random	0.6224	30.03	34.92	35.05
Switchers Policy Tree	1.0342	14.53	23.09	62.38
Switchers random	0.5938	31.55	35.24	33.21

Note: Allocation analysed is DIFFERENT from the one obtained from the training data.