Introduction to EconML

Automated Learning and Intelligence for Causation and Economics (ALICE)

Vasilis Syrgkanis

Microsoft EconML

The MSR ALICE Project

Automated Learning and Intelligence for Causation and Economics

- **Research.** Advance methodological research in econometrics and ML
- **Impact.** Apply Econ + ML methods to industry and societal problems
- **Software.** Develop software tools that reduce barriers to entry

Greg Lewis

Keith Battocchi

Maggie Hei

Miruna Oprescu

Paul Oka

Eleanor Dillon

Vasilis Syrgkanis

EconML is a Python package that applies the power of machine learning techniques to estimate individualized causal responses from observational or experimental data.

Flexible model forms avoid strong assumptions and can estimate personalized responses to treatment

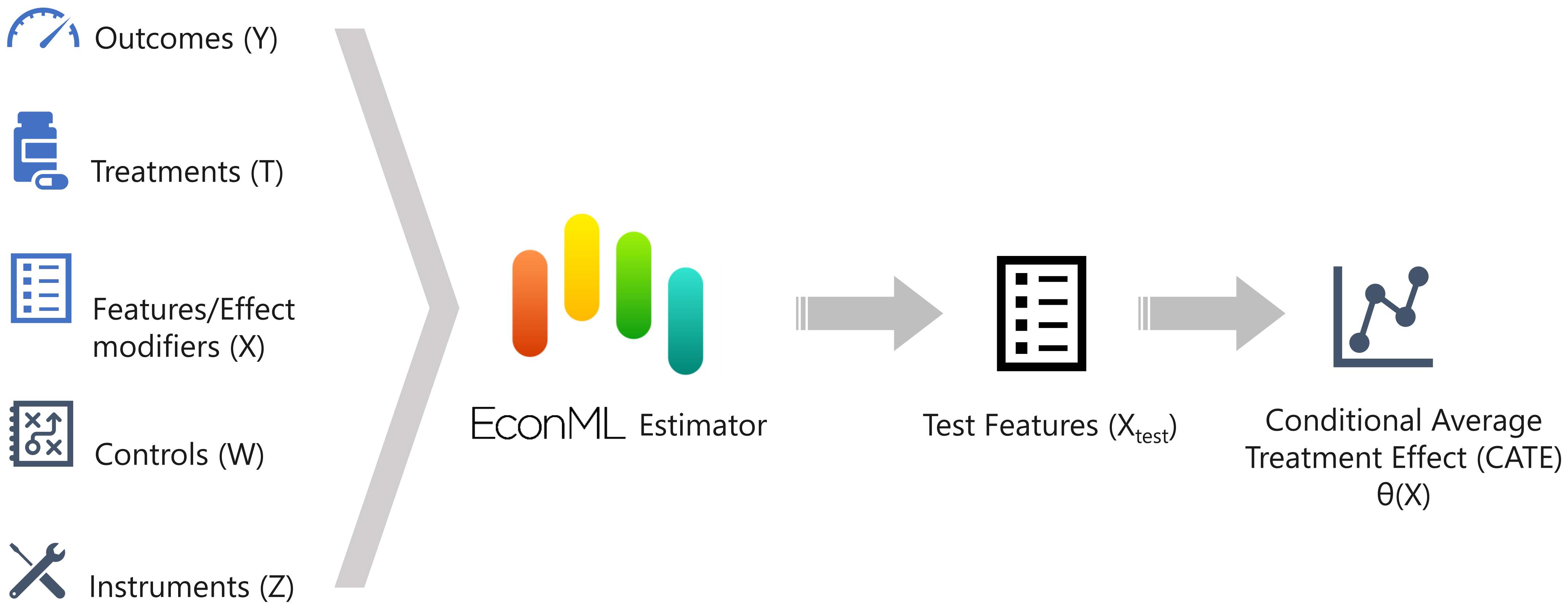
Unified API brings together all the latest advances in causal machine learning and econometrics

Familiar Interface built on standard Python packages make causal analysis quicker and easier for a broad set of users

Quantity of Interest

Conditional Average Treatment Effect (CATE). What is the causal effect on an outcome of interest Y, of switching from a treatment value T0 to treatment value T1 for a sample with features X?

EconML Flow


Outcomes (Y)
Treatments (T)
Features/Effect modifiers (X)
Controls (W)
Instruments (Z)
EconML Estimator
Test Features (X_{test})
Conditional Average Treatment Effect (CATE) $\theta(X)$

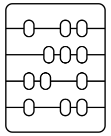
Unified API

Every CATE estimator has a `fit` and `effect` API

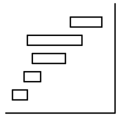
```python
from econml.dml import DML

est = DML() # instantiate CATE estimator
est.fit(y, T, X=X, W=W) # fit the CATE model
est.effect(X, T0=t0, T1=t1) # predict effect for each sample X
```

High-Level Capabilities

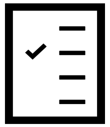
- **Many ML-based methods:** either under conditional exogeneity or with access to instruments
  - DoubleML, Causal Forests, Rlearner, Meta-Learners, DynamicDML, DRLearner, OrthoIV, DRIV, DeepIV, OrthoForests



- **Confidence Intervals:** cutting-edge methods for confidence intervals even when ML is used

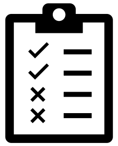


- **Causal Scoring and Causal Cross-Validation:** scoring approaches that target the CATE model

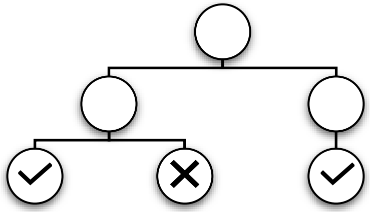


- **Causal Interpretability:** interpreting the CATE with trees and shap values

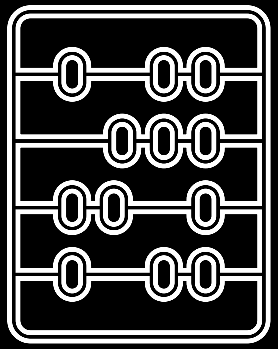
- **Causal Validation:** integration with DoWhy library for sensitivity analysis



- **Policy Recommendations:** recommend interpretable personalized treatment policies



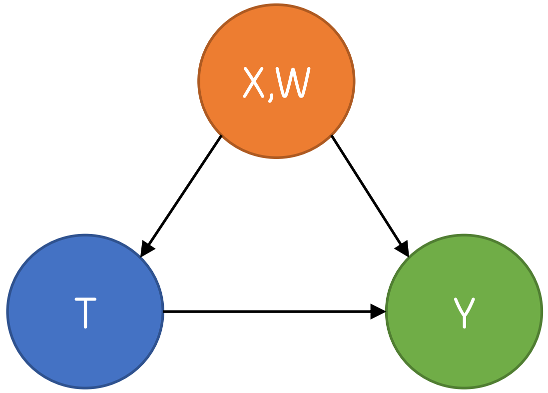
Estimation



Conditional Exogeneity

Confounder: any variable that simultaneously has a direct effect on the treatment and on the outcome

Unconfoundedness: we observe all such confounders



Example Estimator:

Double Machine Learning (Rlearner)

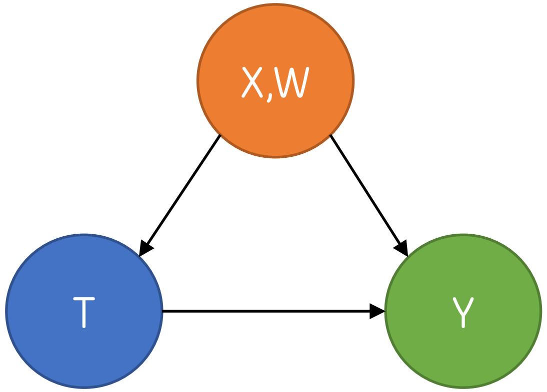
[Nie-Wager17]
[Chernozhukov et al.17]
[Chernozhukov et al.18]
[Foster-Syrgkanis19]

- Cross-fitting: Split your data in two halves
  - Train ML model $\hat{q}$ for $q_0(X, W) \triangleq E[Y|X, W]$ on first, predict on second and calculate residual outcome
    $$\tilde{Y}_i = Y_i - \hat{q}(X_i, W_i)$$
    and vice versa
  - Train ML model $\hat{p}$ for $p_0(X, W) \triangleq E[T\ |X, W]$ on first, predict on second and calculate residual treatment
    $$\tilde{T}_i = T_i - \hat{p}(X_i, W_i)$$
    and vice versa
- Minimize square loss, over CATE model space Θ:

$$\hat{\theta} = \underset{\theta(\cdot)\in\Theta}{\operatorname{argmin}}\, E_n\left[\left(\tilde{Y} - \theta(X)\cdot\tilde{T}\right)^2\right]$$

EconML: Estimation under Exogeneity

```python
from econml.dml import NonParamDML

est = NonParamDML(model_y=RandomForestRegressor(), # Any ML model for E[Y|X,W]
                  model_t=RandomForestRegressor(), # Any ML model for E[T|X,W]
                  model_final=RandomForestRegressor(), # Any ML model for CATE
                  discrete_treatment=False, # categorical or continuous treatment
                  cv=2, # number of cross-fit folds
                  mc_iters=1) # repetitions of cross-fitting for stability

est.fit(y, T, X=X, W=W) # fit the CATE model
est.effect(X, T0=t0, T1=t1) # predict effect for each sample X
```



Personalized effect estimates on test samples

X	point_estimate
0	1.806
1	1.786
2	1.768
3	1.751
4	1.735

Diagnostics on ML models

```python
est.models_y # fitted model_y on each fold
```

```
[[RandomForestRegressor(), RandomForestRegressor()]]
```

```python
est.nuisance_scores_y # out-of-sample scores for model_y
```

```
[[0.6994953731317803, 0.6764212717126863]]
```

Diagnostics on CATE model

```python
est.score_ # in sample goodness-of-fit
```

```
3.5387960870987 28
```

Diagnostics on nuisance quantities

```python
est.residuals_ # calculated residuals (yres, Tres, X, W)
```

econml.dml.DML — econml documentation

EconML: Estimation under Exogeneity

```python
from econml.dml import NonParamDML

est = NonParamDML(model_y=RandomForestRegressor(), # Any ML model for E[Y|X,W]
                  model_t=RandomForestRegressor(), # Any ML model for E[T|X,W]
                  model_final=RandomForestRegressor(), # Any ML model for CATE
                  discrete_treatment=False, # categorical or continuous treatment
                  cv=2, # number of cross-fit folds
                  mc_iters=1) # repetitions of cross-fitting for stability

est.fit(y, T, X=X, W=W) # fit the CATE model
est.effect(X, T0=t0, T1=t1) # predict effect for each sample X
```

```python
from econml.automl import addAutomatedML

AutomatedDML = addAutomatedML(NonParamDML)
est = AutomatedDML(model_y=automl_config_reg, # Azure AutoML for E[Y|X,W]
                   model_t=automl_config_reg, # Azure AutoML for E[Y|X,W]
                   model_final=automl_config_final, # Azure AutoML for CATE
                   cv=2) # number of crossfit folds

est.fit(y, T, X=X, W=W) # fit the CATE model
est.effect(X, T0=t0, T1=t1) # predict effect for each sample X
```

Personalized effect estimates on test samples

X	point_estimate
0	1.806
1	1.786
2	1.768
3	1.751
4	1.735

Diagnostics on ML models

```python
est.models_y # fitted model_y on each fold
```

```
[[RandomForestRegressor(), RandomForestRegressor()]]
```

```python
est.nuisance_scores_y # out-of-sample scores for model_y
```

```
[[0.6994953731317803, 0.6764212717126863]]
```

Diagnostics on CATE model

```python
est.score_ # in sample goodness-of-fit
```

```
3.538796087098728
```

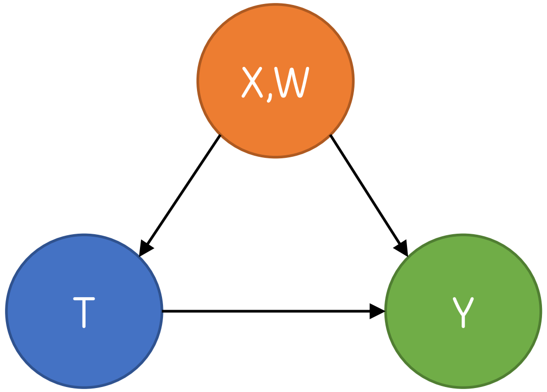
Diagnostics on nuisance quantities

```python
est.residuals_ # calculated residuals (yres, Tres, X, W)
```

ℹ [econml.dml.DML — econml documentation]()

EconML: Estimation under Exogeneity

```python
from econml.dml import NonParamDML

est = NonParamDML(model_y=RandomForestRegressor(), # Any ML model for E[Y|X,W]
                  model_t=RandomForestRegressor(), # Any ML model for E[T|X,W]
                  model_final=RandomForestRegressor(), # Any ML model for CATE
                  discrete_treatment=False, # categorical or continuous treatment
                  cv=2, # number of cross-fit folds
                  mc_iters=1) # repetitions of cross-fitting for stability

est.fit(y, T, X=X, W=W) # fit the CATE model
est.effect(X, T0=t0, T1=t1) # predict effect for each sample X
```



Personalized effect estimates on test samples

X	point_estimate
0	1.806
1	1.786
2	1.768
3	1.751
4	1.735

Diagnostics on ML models

```python
est.models_y # fitted model_y on each fold
```

```
[[RandomForestRegressor(), RandomForestRegressor()]]
```

```python
est.nuisance_scores_y # out-of-sample scores for model_y
```

```
[[0.6994953731317803, 0.6764212717126863]]
```

Diagnostics on CATE model

```python
est.score_ # in sample goodness-of-fit
```

```
3.5387960870987 28
```

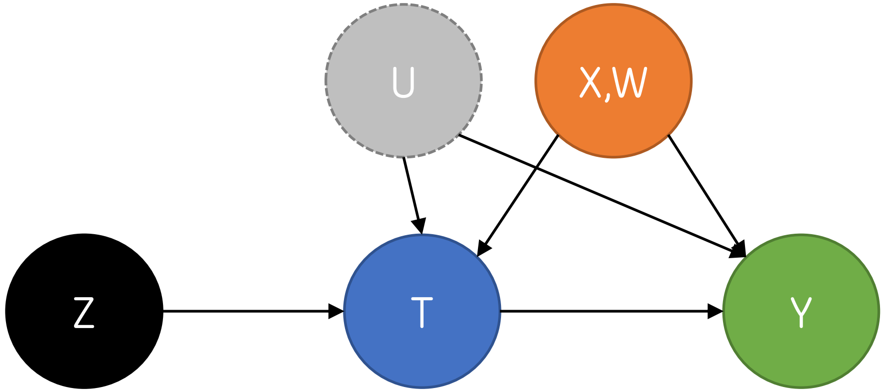
Diagnostics on nuisance quantities

```python
est.residuals_ # calculated residuals (yres, Tres, X, W)
```

[econml.dml.DML — econml documentation]

EconML: Estimation with Instruments

```
from econml.iv.dml import OrthoIV

est = OrthoIV(model_y_xw=RandomForestRegressor(), # ML model for E[Y|X,W]
              model_t_xw=RandomForestRegressor(), # ML model for E[T|X,W]
              model_z_xw=RandomForestRegressor(), # ML model for E[Z|X,W]
              discrete_treatment=False, # categorical/continuous treatment
              discrete_instrument=False, # categorical/continuous instrument
              cv=2, # number of cross-fit folds
              mc_iters=1) # repetitions of cross-fitting for stability

est.fit(y, T, Z=Z, X=X, W=W) # fit the CATE model
est.effect(X, T0=t0, T1=t1) # predict effect for each sample X
```



Personalized effect estimates on test samples

X	point_estimate
0	1.806
1	1.786
2	1.768
3	1.751
4	1.735

Diagnostics on ML models

```
est.models_y_xw # fitted model_y_xw on each fold
```

```
[[RandomForestRegressor(), RandomForestRegressor()]]
```

```
est.nuisance_scores_y_xw # out-of-sample scores for model_y_xw
```

```
[[0.6994953731317803, 0.6764212717126863]]
```

Diagnostics on CATE model

```
est.score_ # in sample goodness-of-fit
```

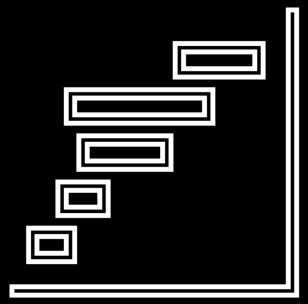
```
3.538796087098728
```

Diagnostics on nuisance quantities

```
est.residuals_ # calculated residuals (yres, Tres, Zres, X, W, Z)
```

[OrthoIV— econml documentation]

Inference

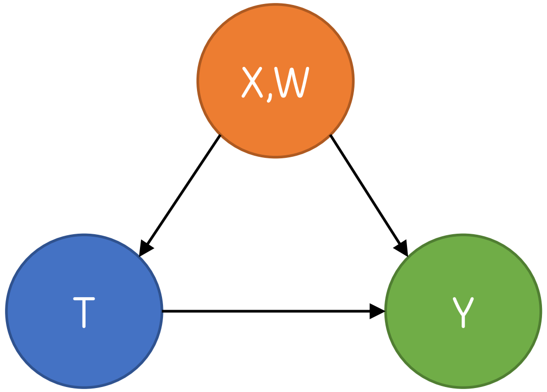


EconML: Estimation under Exogeneity

```
from econml.dml import NonParamDML

est = NonParamDML(model_y=RandomForestRegressor(), # Any ML model for E[Y|X,W]
                  model_t=RandomForestRegressor(), # Any ML model for E[T|X,W]
                  model_final=RandomForestRegressor(), # Any ML model for CATE
                  discrete_treatment=False, # categorical or continuous treatment
                  cv=2, # number of cross-fit folds
                  mc_iters=1) # repetitions of cross-fitting for stability

est.fit(y, T, X=X, W=W) # fit the CATE model
est.effect(X, T0=t0, T1=t1) # predict effect for each sample X
```

Personalized effect estimates on test samples

X	point_estimate
0	1.806
1	1.786
2	1.768
3	1.751
4	1.735



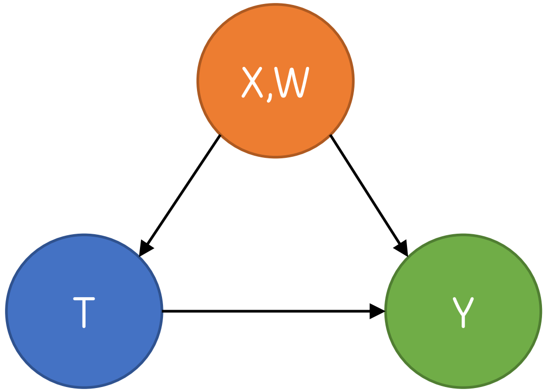
Inference — econml documentation

EconML: Generic Bootstrap Inference

```python
from econml.dml import NonParamDML

est = NonParamDML(model_y=RandomForestRegressor(), # Any ML model for E[Y|X,W]
                  model_t=RandomForestRegressor(), # Any ML model for E[T|X,W]
                  model_final=RandomForestRegressor(), # Any ML model for CATE
                  discrete_treatment=False, # categorical or continuous treatment
                  cv=2, # number of cross-fit folds
                  mc_iters=1) # repetitions of cross-fitting for stability

est.fit(y, T, X=X, W=W, inference='bootstrap') # enable bootstrap intervals
est.effect_inference(X, T0=t0, T1=t1) # predict effect for each sample X
```

Personalized effect estimates on test samples

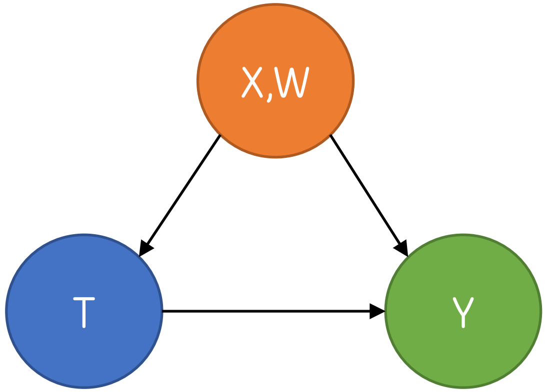
X	point_estimate	stderr	zstat	pvalue	ci_lower	ci_upper
0	1.806	0.127	14.275	0.0	1.598	2.015
1	1.786	0.120	14.934	0.0	1.589	1.983
2	1.768	0.113	15.653	0.0	1.582	1.953
3	1.751	0.106	16.439	0.0	1.575	1.926
4	1.735	0.100	17.295	0.0	1.570	1.900



Inference — econml documentation

EconML: Generic Bootstrap Inference

```python
from econml.dml import NonParamDML

est = NonParamDML(model_y=RandomForestRegressor(), # Any ML model for E[Y|X,W]
                  model_t=RandomForestRegressor(), # Any ML model for E[T|X,W]
                  model_final=RandomForestRegressor(), # Any ML model for CATE
                  discrete_treatment=False, # categorical or continuous treatment
                  cv=2, # number of cross-fit folds
                  mc_iters=1) # repetitions of cross-fitting for stability

est.fit(y, T, X=X, W=W, inference='bootstrap') # enable bootstrap intervals
est.effect_inference(X, T0=t0, T1=t1) # predict effect for each sample X
est.ate_inference(X_test) # inference on average effect of test population
```



Personalized effect estimates on test samples

X	point_estimate	stderr	zstat	pvalue	ci_lower	ci_upper
0	1.806	0.127	14.275	0.0	1.598	2.015
1	1.786	0.120	14.934	0.0	1.589	1.983
2	1.768	0.113	15.653	0.0	1.582	1.953
3	1.751	0.106	16.439	0.0	1.575	1.926
4	1.735	0.100	17.295	0.0	1.570	1.900

Inference on average effects over test samples

Uncertainty of Mean Point Estimate

mean_point	stderr_mean	zstat	pvalue	ci_mean_lower	ci_mean_upper
1.378	0.26	5.303	0.0	0.95	1.805

Distribution of Point Estimate

std_point	pct_point_lower	pct_point_upper
0.607	0.437	2.331

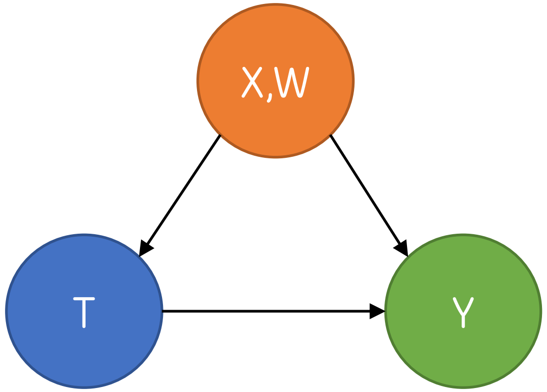
Total Variance of Point Estimate

stderr_point	ci_point_lower	ci_point_upper
0.661	0.386	2.511

Inference — econml documentation

EconML: Generic Bootstrap Inference

```python
from econml.dml import NonParamDML

est = NonParamDML(model_y=RandomForestRegressor(), # Any ML model for E[Y|X,W]
                  model_t=RandomForestRegressor(), # Any ML model for E[T|X,W]
                  model_final=RandomForestRegressor(), # Any ML model for CATE
                  discrete_treatment=False, # categorical or continuous treatment
                  cv=2, # number of cross-fit folds
                  mc_iters=1) # repetitions of cross-fitting for stability

# customize bootstrap parameters
from econml.inference import BootstrapInference
est.fit(y, T, X=X, W=W, inference=BootstrapInference(n_bootstrap_samples=10,
                                                     bootstrap_type='normal'))
```



Personalized effect estimates on test samples

X	point_estimate	stderr	zstat	pvalue	ci_lower	ci_upper
0	1.806	0.127	14.275	0.0	1.598	2.015
1	1.786	0.120	14.934	0.0	1.589	1.983
2	1.768	0.113	15.653	0.0	1.582	1.953
3	1.751	0.106	16.439	0.0	1.575	1.926
4	1.735	0.100	17.295	0.0	1.570	1.900

Inference on average effects over test samples

Uncertainty of Mean Point Estimate

mean_point	stderr_mean	zstat	pvalue	ci_mean_lower	ci_mean_upper
1.378	0.26	5.303	0.0	0.95	1.805

Distribution of Point Estimate

std_point	pct_point_lower	pct_point_upper
0.607	0.437	2.331

Total Variance of Point Estimate

stderr_point	ci_point_lower	ci_point_upper
0.661	0.386	2.511

ℹ [Inference — econml documentation]

EconML: Tailored Valid Inference

Linear CATE
$\langle\theta, \phi(X)\rangle$

```
from econml.dml import LinearDML

est = LinearDML(featurizer=PolynomialFeatures(degree=2)) # featurizer for phi(X)
est.fit(y, T, X=X, W=W) # inference enabled by default

# Inference with heteroskedasticity robust OLS standard errors
est.summary()
est.effect_inference(X_test, T0=t0, T1=t1)
```

high-dimensional linear CATE
$\langle\theta, \phi(X)\rangle$

```
from econml.dml import SparseLinearDML

# high-dimensional featurizer for phi(X)
est = SparseLinearDML(featurizer=PolynomialFeatures(degree=5))
est.fit(y, T, X=X, W=W) # inference enabled by default

# Inference with the debiased Lasso
est.summary()
est.effect_inference(X_test, T0=t0, T1=t1)
```

Forest CATE

```
from econml.dml import CausalForestDML

# fit forest based model for theta(X)
est = CausalForestDML(criterion='mse', n_estimators=1000, ...)
est.tune(y, T, X=X, W=W).fit(y, T, X=X, W=W) # inference enabled by default

# Inference with bootstrap-of-little-bags for forests
est.summary()
est.effect_inference(X_test, T0=t0, T1=t1)
```

Inference — econml documentation

Inference on parameters of CATE model

	point_estimate	stderr	zstat	pvalue	ci_lower	ci_upper
X0	-1.46	0.674	-2.165	0.03	-2.57	-0.351
X0^2	7.341	0.761	9.648	0.0	6.089	8.592

CATE Intercept Results

	point_estimate	stderr	zstat	pvalue	ci_lower	ci_upper
cate_intercept	1.809	0.119	15.243	0.0	1.614	2.004

Inference on personalized effects on test samples

X	point_estimate	stderr	zstat	pvalue	ci_lower	ci_upper
0	1.806	0.127	14.275	0.0	1.598	2.015
1	1.786	0.120	14.934	0.0	1.589	1.983
2	1.768	0.113	15.653	0.0	1.582	1.953
3	1.751	0.106	16.439	0.0	1.575	1.926
4	1.735	0.100	17.295	0.0	1.570	1.900

Inference on average effects on training samples

Doubly Robust ATE on Training Data Results

	point_estimate	stderr	zstat	pvalue	ci_lower	ci_upper
ATE	3.158	0.082	38.551	0.0	3.023	3.292

Doubly Robust ATT(T=0) on Training Data Results

	point_estimate	stderr	zstat	pvalue	ci_lower	ci_upper
ATT	3.1	0.096	32.322	0.0	2.942	3.258

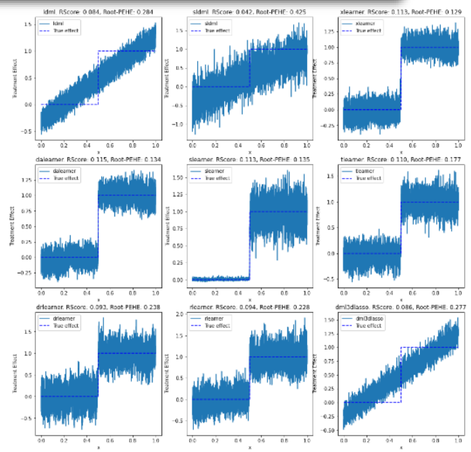
Scoring

EconML: Scoring

- Multitude of approaches for CATE estimation to select from

```python
# A multitude of possible approaches for CATE estimation under conditional exogeneity
models = [('ldml', LinearDML(model_y=reg(), model_t=clf(), discrete_treatment=True)),
          ('sldml', SparseLinearDML(model_y=reg(), model_t=clf(), discrete_treatment=True,
                                    featurizer=PolynomialFeatures(degree=2, include_bias=False))),
          ('xlearner', XLearner(models=reg(), cate_models=reg(), propensity_model=clf())),
          ('dalearner', DomainAdaptationLearner(models=reg(), final_models=reg(), propensity_model=clf())),
          ('slearner', SLearner(overall_model=reg())),
          ('tlearner', TLearner(models=reg())),
          ('drlearner', DRLearner(model_propensity=clf(), model_regression=reg(), model_final=reg())),
          ('rlearner', NonParamDML(model_y=reg(), model_t=clf(), model_final=reg(), discrete_treatment=True)),
          ('dml3dlasso', DML(model_y=reg(), model_t=clf(), model_final=LassoCV(), discrete_treatment=True,
                             featurizer=PolynomialFeatures(degree=3)))]
```

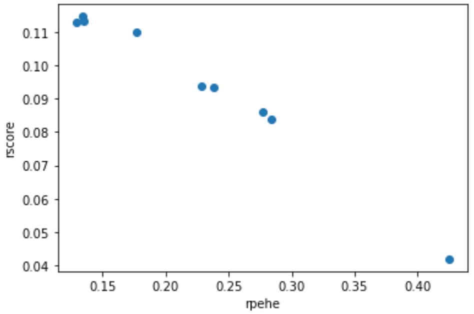
Qualitatively different performance



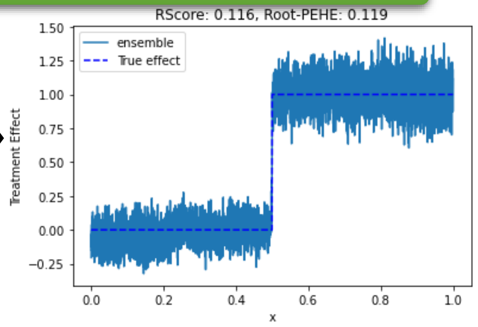
- A universal causal scorer: RScorer

```python
from econml.score import RScorer

# Causal score actually needs fitting on the validation set!
scorer = RScorer(model_y=reg(), model_t=clf()).fit(Y_val, T_val, X=X_val)
# Then we can evaluate every trained CATE model
rscore = [scorer.score(mdl) for _, mdl in models]
```

RScore correlates with ideal score



Choose model with highest RScore

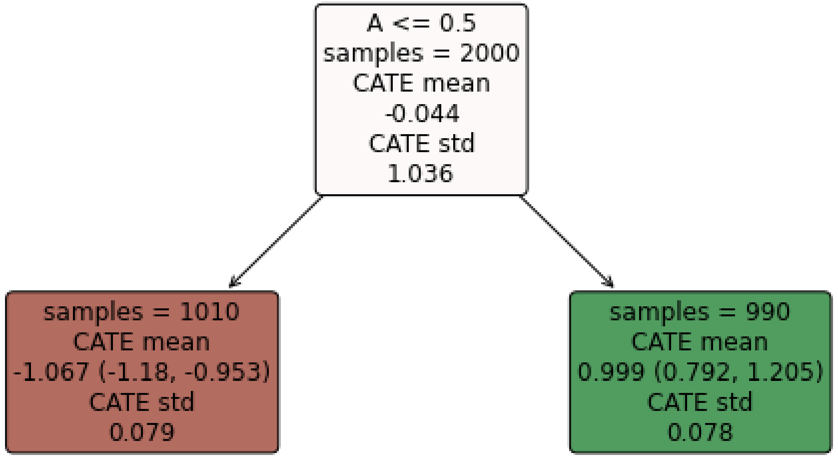


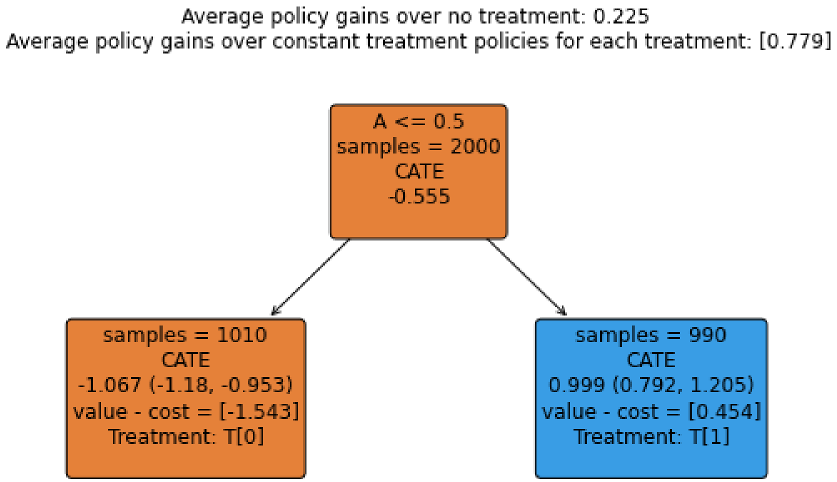
[econml.score.RScorer — econml documentation]()

Interpretation

EconML: Interpretability with Trees

```python
from econml.cate_interpreter import SingleTreeCateInterpreter

# interpret any CATE model's behavior on a set of samples as a single tree
intrp = SingleTreeCateInterpreter(max_depth=1)
intrp.interpret(est, X)

intrp.export_graphviz(out_file='cate_tree.dot') # export to a dot file
intrp.plot() # plot with matplotlib
```





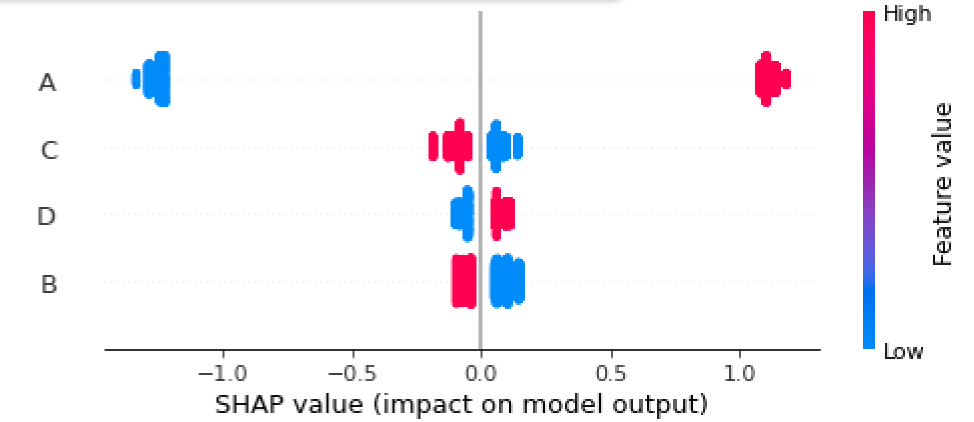

```python
from econml.cate_interpreter import SingleTreePolicyInterpreter

# make tree-based policy recommendations from CATE model
intrp = SingleTreePolicyInterpreter(risk_level=0.05, max_depth=1)
intrp.interpret(est, X, sample_treatment_costs=0.2)

intrp.export_graphviz(out_file='policy_tree.dot') # export to a dot file
intrp.plot() # plot with matplotlib
```

Interpretability — econml documentation

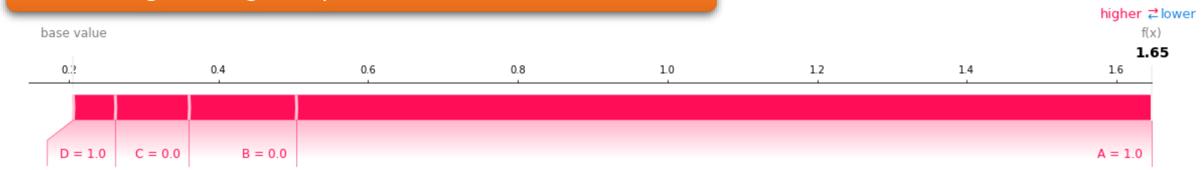
EconML: Interpretability with SHAP

```python
import shap

# shap values for CATE predictions of any CATE estimator
shap_values = est.shap_values(X)
# effect heterogeneity feature importances with summary plot
shap.summary_plot(shap_values['Y0']['T0'])
# explain the heterogeneity of effect of any single sample
shap.force_plot(shap_values['Y0']['T0'][sample_id])
```

Summary plot for heterogeneity feature importances



Attributing heterogeneity of individual CATE values



[slundberg/shap: explain any machine learning model]

[Interpretability — econml documentation]

Validation and Sensitivity

EconML: Validation with DoWhy

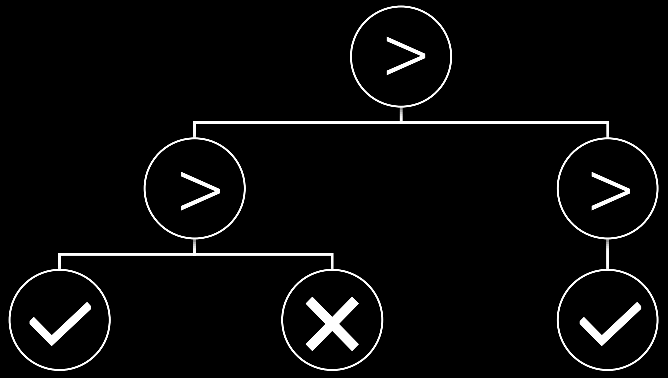
- Sensitivity analysis with DoWhy: sensitivity to causal assumptions

```python
import dowhy

# enabled dowhy capabilities by using the dowhy wrapper
est = est.dowhy
est.fit(y, T, Z=Z, X=X, W=W)

# econml API is preserved
est.summary()
est.effect(X)

# dowhy capabilities are now also available: sensitivity analysis
est.refute_estimate(method_name="add_unobserved_common_cause",
                    effect_strength_on_treatment=0.05, effect_strength_on_outcome=0.5)
```

```
Refute: Add an Unobserved Common Cause
Estimated effect:2.484810103996173
New effect:2.7511514869150684
```

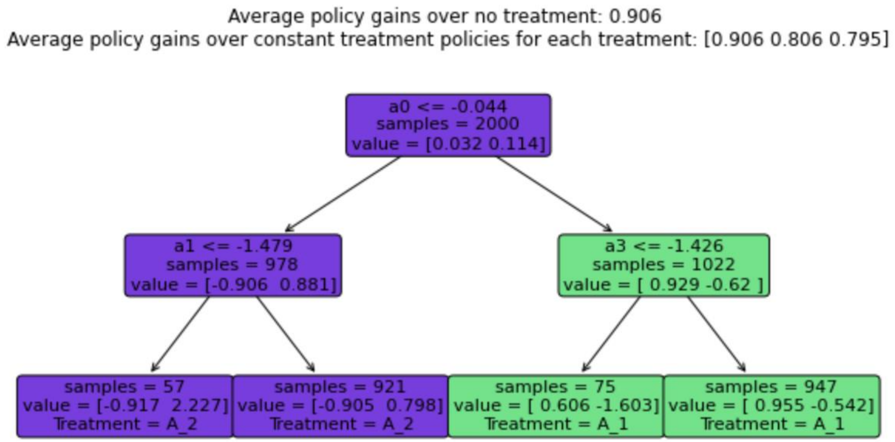
microsoft/dowhy: DoWhy is a Python library for causal inference

Policy Learning



EconML: Policy Learning

```python
from econml.policy import DRPolicyTree, DRPolicyForest

est = DRPolicyTree(max_depth=2, min_impurity_decrease=0.01, honest=True, ...)
est.fit(y, T, X=X, W=W)
est.plot() # plot tree with matplotlib
est.feature_importances_ # feature importances
est.predict(X) # produce recommended treatment for each sample

est = DRPolicyForest(n_estimators=100, ...)
est.fit(y, T, X=X, W=W)
est.predict(X) # produce recommended treatment for each sample
```

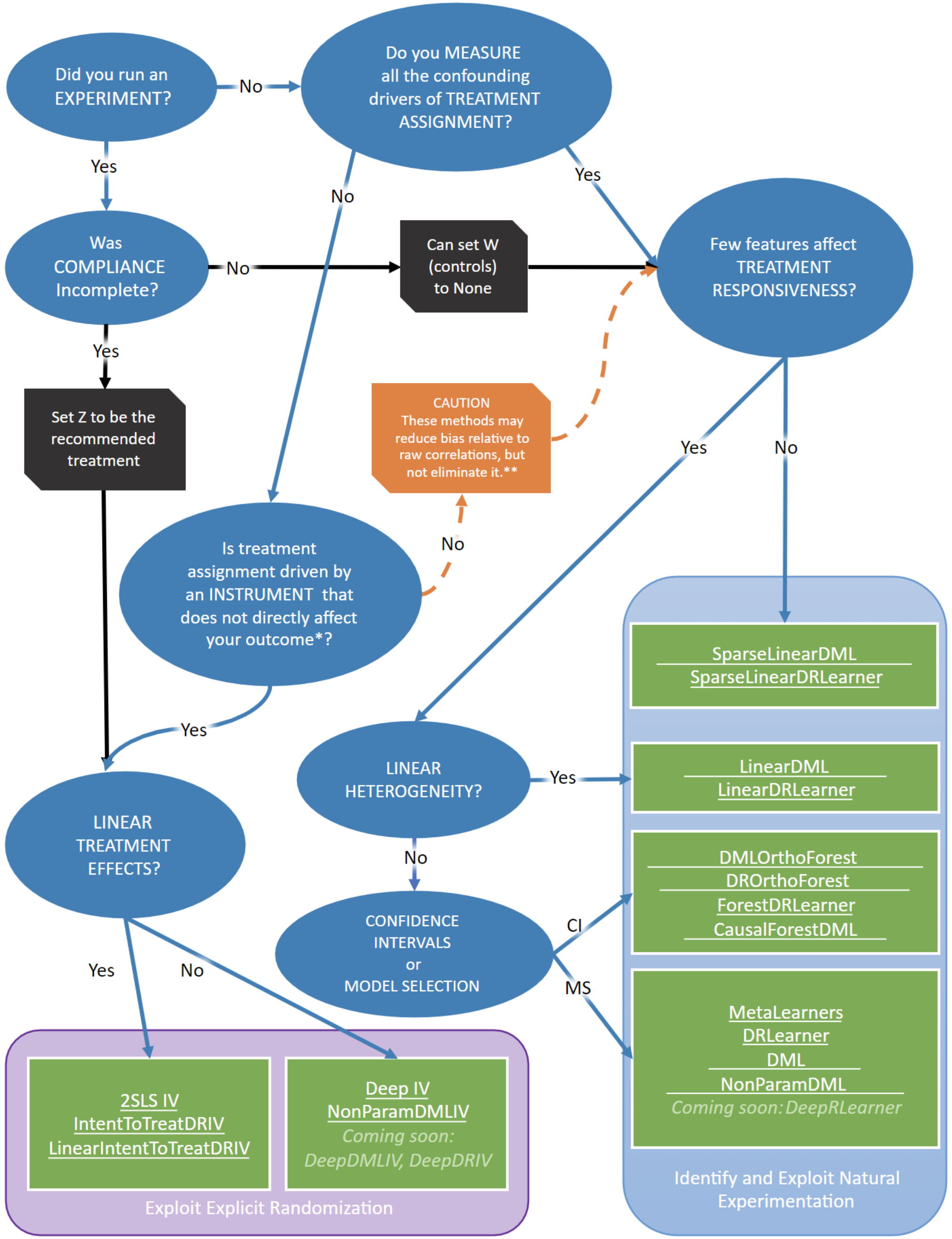


[econml.policy.DRPolicyForest — econml documentation]

[econml.policy.DRPolicyTree — econml documentation]

Overview

Estimator	Treatment Type	Requires Instrument	Delivers Conf. Intervals	Linear Treatment	Linear Heterogeneity	Mulitple Outcomes	Multiple Treatments	High-Dimensional Features
NonparametricTwoStageLeastSquares	Any	Yes		Yes	Assumed	Yes	Yes	
DeepIV	Any	Yes				Yes	Yes	
SparseLinearDML	Any		Yes	Yes	Assumed	Yes	Yes	Yes
SparseLinearDRLearner	Categorical		Yes		Projected		Yes	Yes
LinearDML	Any		Yes	Yes	Assumed	Yes	Yes	
LinearDRLearner	Categorical		Yes		Projected		Yes	
CausalForestDML	Any		Yes	Yes		Yes	Yes	Yes
ForestDRLearner	Categorical		Yes				Yes	Yes
DMLOrthoForest	Any		Yes	Yes			Yes	Yes
DROrthoForest	Categorical		Yes				Yes	Yes
metalearners	Categorical					Yes	Yes	Yes
DRLearner	Categorical						Yes	Yes
DML	Any			Yes	Assumed	Yes	Yes	Yes
NonParamDML	1-d/Binary			Yes		Yes		Yes
OrthoIV	Any	Yes	Yes	Yes	Assumed	Yes	Yes	
DRIV	1-d/Binary	Yes	Yes	Yes				Yes


Did you run an EXPERIMENT?
No
Do you MEASURE all the confounding drivers of TREATMENT ASSIGNMENT?
Yes
No
Yes
Was COMPLIANCE Incomplete?
No
Can set W (controls) to None
Few features affect TREATMENT RESPONSIVENESS?
Yes
Set Z to be the recommended treatment
CAUTION
These methods may reduce bias relative to raw correlations, but not eliminate it.**
Yes
No
No
Is treatment assignment driven by an INSTRUMENT that does not directly affect your outcome*?
SparseLinearDML
SparseLinearDRLearner
Yes
LINEAR HETEROGENEITY?
Yes
LinearDML
LinearDRLearner
LINEAR TREATMENT EFFECTS?
No
DMLOrthoForest
DROrthoForest
ForestDRLearner
CausalForestDML
CONFIDENCE INTERVALS or MODEL SELECTION
CI
MS
Yes
No
MetaLearners
DRLearner
DML
NonParamDML
Coming soon: DeepRLearner
2SLS IV
IntentToTreatDRIV
LinearIntentToTreatDRIV
Deep IV
NonParamDMLIV
Coming soon: DeepDMLIV, DeepDRIV
Exploit Explicit Randomization
Identify and Exploit Natural Experimentation

User Guide

EconML python library for ML Estimation of Heterogeneous Treatment Effects
https://github.com/microsoft/EconML
`pip install econml`

Notebook with samples in presentation

ALICE (Automated Learning and Intelligence for Causation and Economics) project:
https://www.microsoft.com/en-us/research/project/alice/

Academic References

1. G Lewis, V Syrgkanis. **Double/Debiased Machine Learning for Dynamic Treatment Effects via g-Estimation.** Arxiv (2021).

2. Athey, Susan, and Stefan Wager. **Policy learning with observational data.** Econometrica 89.1 (2021): 133-161.

3. X Nie, S Wager. **Quasi-Oracle Estimation of Heterogeneous Treatment Effects.** *Biometrika*, 2020

4. V. Syrgkanis, V. Lei, M. Oprescu, M. Hei, K. Battocchi, G. Lewis. **Machine Learning Estimation of Heterogeneous Treatment Effects with Instruments.** *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019 **(Spotlight Presentation)**

5. D. Foster, V. Syrgkanis. **Orthogonal Statistical Learning.** *Proceedings of the 32nd Annual Conference on Learning Theory (COLT)*, 2019 **(Best Paper Award)**

6. M. Oprescu, V. Syrgkanis and Z. S. Wu. **Orthogonal Random Forest for Causal Inference.** *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

7. S. Künzel, J. Sekhon, J. Bickel and B. Yu. **Metalearners for estimating heterogeneous treatment effects using machine learning.** *Proceedings of the national academy of sciences, 116(10), 4156-4165*, 2019.

8. S. Athey, J. Tibshirani, S. Wager. **Generalized random forests.** *Annals of Statistics, 47, no. 2, 1148--1178*, 2019.

9. V Semenova, V Chernozhukov. Debiased machine learning of conditional average treatment effects and other causal functions. The Econometrics Journal, 2021

10. V. Chernozhukov, D. Nekipelov, V. Semenova, V. Syrgkanis. **Plug-in Regularized Estimation of High-Dimensional Parameters in Nonlinear Semiparametric Models.** *Arxiv preprint arxiv:1806.04823*, 2018.

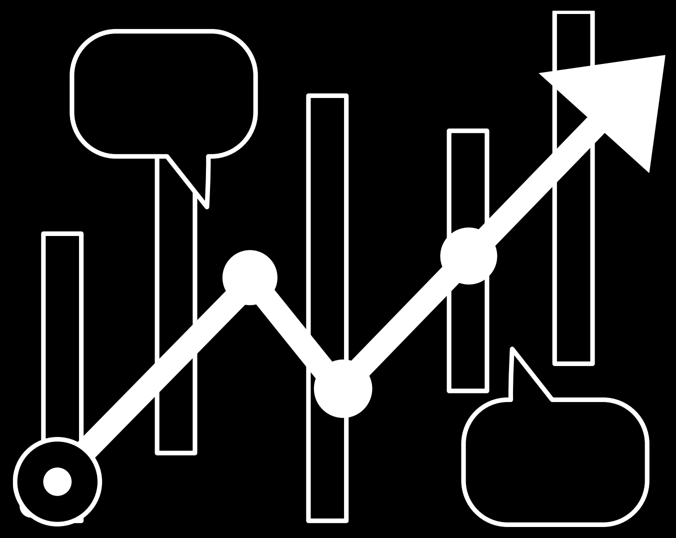
11. S. Wager, S. Athey. **Estimation and Inference of Heterogeneous Treatment Effects using Random Forests.** *Journal of the American Statistical Association, 113:523, 1228-1242*, 2018.

12. Jason Hartford, Greg Lewis, Kevin Leyton-Brown, and Matt Taddy. **Deep IV: A flexible approach for counterfactual prediction.** *Proceedings of the 34th International Conference on Machine Learning, ICML'17*, 2017.

13. V. Chernozhukov, D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, and a. W. Newey. **Double Machine Learning for Treatment and Causal Parameters.** *ArXiv preprint arXiv:1608.00060*, 2016.

14. Dudik, M., Erhan, D., Langford, J., & Li, L. **Doubly robust policy evaluation and optimization.** Statistical Science, 29(4), 485-511, 2014.

Case Studies



1. Customer Segmentation at TripAdvisor with Recommendation A/B Tests

2. Long-Term Return-on-Investment at Microsoft via Short-Term Proxies