Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

DoubleML - Double Machine Learning in R

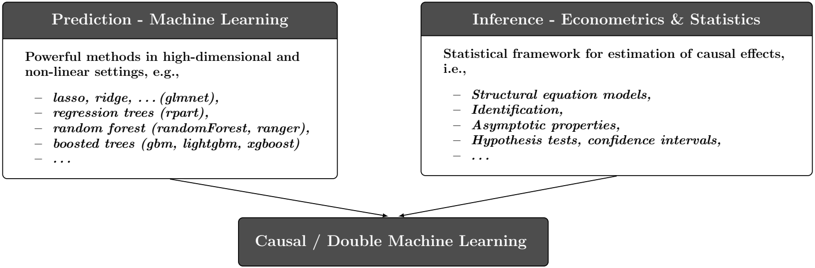
Philipp Bach, Malte S. Kurz

philipp.bach@uni-hamburg.de
malte.simon.kurz@uni-hamburg.de

UseR! 5-9 July, 2021

Introduction to Double Machine Learning

What is Double Machine Learning (DML)?



- **Result / output** from the DML framework:
  - Estimate of a causal effect or structural parameter with valid confidence intervals → statistical tests for parameter(s) of interest
  - Good statistical properties ($\sqrt{N}$ rate of convergence; unbiased; approximately Gaussian)

Causal Machine Learning: Examples

- Evaluation of an intervention in randomized controlled trials or observational studies, e.g., A/B testing, clinical studies, program evaluation
- General: What is the **effect** of a certain **treatment** on a relevant **outcome** variable?

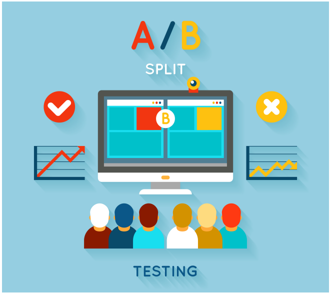


Image Source: Freepik (http://www.freepik.com) / Designed by macrovector

Example: Partially Linear Regression

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- Partially linear regression (PLR) model

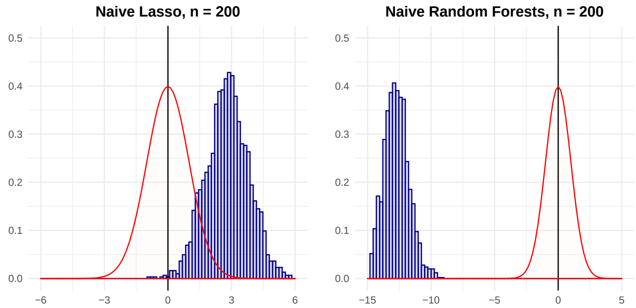
$$Y = D\theta_0 + g_0(X) + \zeta, \qquad \mathbb{E}[\zeta|D, X] = 0,$$

with potentially non-linear function $g_0()$ and

  - Outcome variable Y
  - Policy or treatment variable of interest D
  - High-dimensional vector of confounding covariates $X = (X_1, \dots, X_p)$

Problems in Naive Approaches

- Failure of naive approaches: **Regularization bias**, e.g.,
  - Naive variable selection, for example based on the lasso
  - Naive plug-in predictions, for example from random forests



What is Double Machine Learning (DML)?

- **Double/debiased machine learning (DML)** (Chernozhukov et al. 2018): General framework for estimation of treatment effects based on machine learning (ML)
- The parameter of interest, θ_0, is identified as the solution to a moment condition

$$\mathbb{E}\left[\psi(W;\theta_0,\eta_0)\right] = 0,$$

with score function $\psi(\cdot)$, i.i.d. data W and nuisance term η.

- **Key ingredients** of the DML approach
  1. Neyman orthogonality,
  2. High-quality machine learning estimators,
  3. Sample splitting.

The Key Ingredients of DML

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

1. Neyman Orthogonality

- An essential property of the score is **Neyman orthogonality**

$$\partial_\eta \mathbb{E}[\psi(W;\theta_0,\eta)]\big|_{\eta=\eta_0} = 0.$$

- The moment condition identifying θ_0 is *insenstive to small pertubations of η around η_0*. $\Rightarrow$ Estimation *immunized* against first order biases from replacing η_0 by ML estimator $\widehat{\eta}_0$.

- PLR example: Inclusion of first-stage regression

$$D = m(X) + V,$$

leads to the Neyman-orthogonal score

$$\psi(W;\theta,\eta) := (Y - g(X) - \theta D)\,(D - m(X)),$$

with $\eta = \{g, m\}$.

The Key Ingredients of DML

2. High-Quality Machine Learning Estimators

The nuisance parameters are estimated with high-quality machine learning methods, i.e., η_0 is estimated at a sufficiently fast rate of convergence.

- Different structural assumptions on η_0 lead to the use of different machine-learning tools for estimating η_0

3. Sample Splitting

To avoid the biases arising from overfitting, a form of **sample splitting** is used at the stage of producing the estimator of the main parameter θ_0.

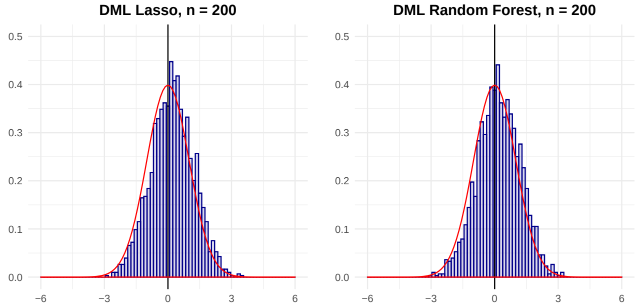
- Fit ML models on *train sample*, generate predictions on *test sample*. Swap the roles to gain full efficiency (cross-fitting). Plug in predictions into the score and solve for θ_0

The Double Machine Learning Framework

- Under regularity conditions, it can be shown that

$$\sqrt{N}(\tilde{\theta}_0 - \theta_0) \rightsquigarrow N(0, \sigma^2).$$

- For more details, see Chernozhukov et al. (2018) and the **package vignette** (Bach et al. 2021b)



The R Package DoubleML

The R Package DoubleML: Building Principles

DoubleML - Building Principles

Key ingredient	**Implementation**
1. Orthogonal score	Object-oriented implementation with `R6`; exploit common structure centered around a (linear) score function $\psi(\cdot)$
2. High-quality ML	State-of-the art ML prediction & tuning methods provided by `mlr3` ecosystem (meta packages)
3. Sample splitting	Built-in resampling schemes of `mlr3`

The R Package DoubleML: Main Dependencies and Installation

Dependencies

mlr3

R6

data.table

mlr3learners

mlr3tuning

Installation

- **Latest** CRAN **release**

```
$ install.packages('DoubleML')
```

- **Development version** from GitHub

```
$ remotes::install_github('DoubleML/doubleml-for-r')
```

Why an Object-Orientated Implementation?

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

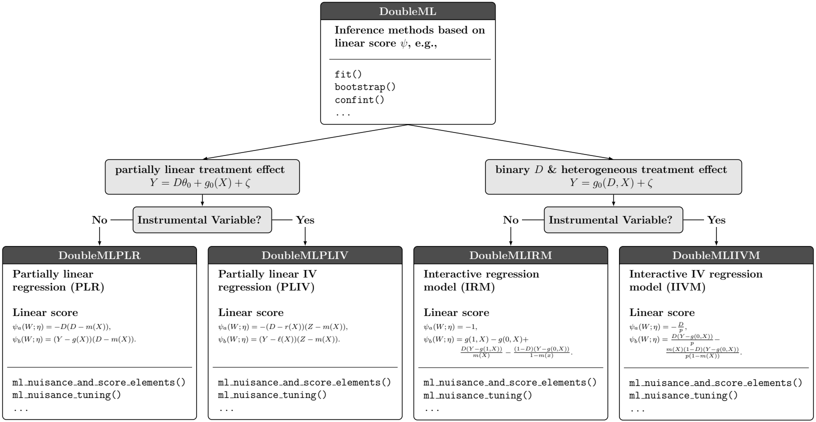
- Given the components ($\psi^a(\cdot)$ & $\psi^b(\cdot)$) of a linear Neyman orthogonal score function $\psi(\cdot)$, a **general implementation** is possible for
  - The estimation of the **orthogonal parameters**
  - The computation of the **score** $\psi(W; \theta, \eta)$
  - The estimation of **standard errors**
  - The computation of **confidence intervals**
  - A **multiplier bootstrap** procedure for simultaneous inference
- The **sample splitting** can be implemented in general as well

$\rightarrow$ Implemented in the **abstract base class** `DoubleML`

- The **score components** and the estimation of the **nuisance models** have to be implemented **model-specifically**

$\rightarrow$ Implemented in **model-specific classes** inherited from `DoubleML`

Class Structure and Causal Models



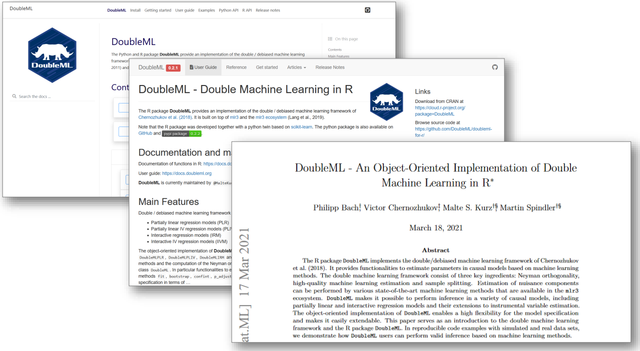
Advantages of the Object-Orientation

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- `DoubleML` gives the user a **high flexibility** with regard to the specification of DML models:
  - Choice of ML methods for approximating the nuisance functions
  - Different resampling schemes (repeated cross-fitting)
  - DML algorithms DML1 and DML2
  - Different Neyman orthogonal score functions
- `DoubleML` can be **easily extended**
  - New model classes with appropriate Neyman orthogonal score function can be inherited from `DoubleML`
  - The package features `callables` as score functions which makes it easy to extend existing model classes
  - The resampling schemes are customizable in a flexible way

Online Resources

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG



- Documentation and User Guide available at

  https://docs.doubleml.org

- Package vignette available via **arXiv:2103.09603**

Thank you very much for your attention!

philipp.bach@uni-hamburg.de
malte.simon.kurz@uni-hamburg.de

In case you like our project, we appreciate stars on GitHub :-)

`https://github.com/DoubleML/doubleml-for-r`

DoubleML: Package Papers

- **Papers / Vignette**

P. Bach, V. Chernozhukov, M. S. Kurz, and M. Spindler (2021b), *DoubleML – An Object-Oriented Implementation of Double Machine Learning in R*, arXiv:2103.09603

P. Bach, V. Chernozhukov, M. S. Kurz, and M. Spindler (2021a), *DoubleML – An Object-Oriented Implementation of Double Machine Learning in Python*, arXiv:2104.03220

References

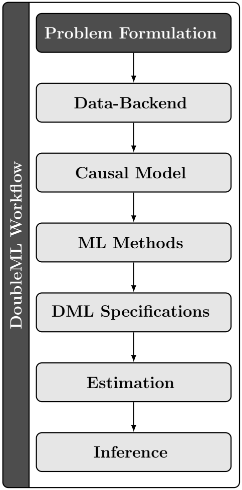
Bach, P., V. Chernozhukov, M. S. Kurz, and M. Spindler (2021a), *DoubleML – An Object-Oriented Implementation of Double Machine Learning in Python*, arXiv:2104.03220.

— (2021b), *DoubleML – An Object-Oriented Implementation of Double Machine Learning in R*, arXiv:2103.09603.

Chernozhukov, V., D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, and J. Robins (2018), "Double/debiased machine learning for treatment and structural parameters", *The Econometrics Journal* 21(1), pp. C1–C68.

Appendix: DoubleML Workflow

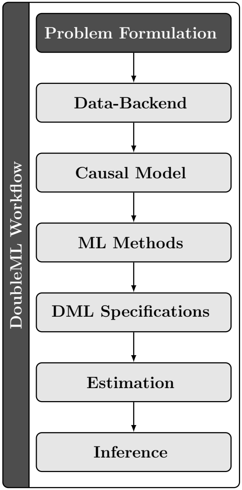
DoubleML Workflow: 0. Problem Formulation



Description of the Case Study and Data I

- 401(k) plans are pension accounts sponsored by employers
- **Estimate the effect of 401(k) eligibility and participation on accumulated assets**
- Problems: **Saver heterogeneity** and the fact that the **decision to enroll** in a 401(k) **is non-random**
- **Conventional estimates** that do not account for saver heterogeneity and endogeneity of participation **will be biased**

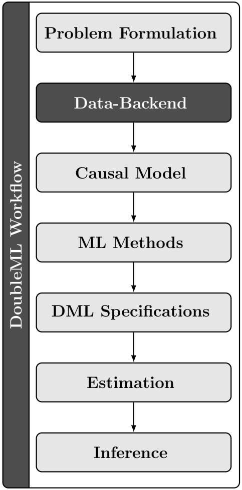
DoubleML Workflow: 0. Problem Formulation



Description of the Case Study and Data II

- **Eligibility** for enrolling in a 401(k) plan might be taken as **exogenous after conditioning** on a few observables of which the most important may be income
- **The basic idea**: Around the time 401(k)'s initially became available, people were unlikely to be basing their employment decisions on whether an employer offered a 401(k) but would instead focus on income and other aspects of the job
- The data consist of 9,915 observations at the household level drawn from the 1991 Survey of Income and Program Participation (SIPP)
- **Outcome variable: Net financial assets**

DoubleML Workflow: 1. Data-Backend



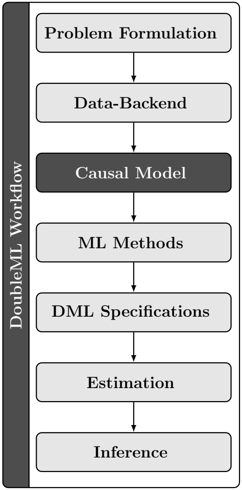
- **DoubleMLData** from a data.table or data.frame

```r
library(DoubleML)
data = fetch_401k(return_type='data.table')
# Construct DoubleMLData object from data.table
dml_data = DoubleMLData$new(data, y_col='net_tfa', d_cols='e401',
                            x_cols=c('age', 'inc', 'educ', 'fsize',
                                     'marr', 'twoearn', 'db', 'pira',
                                     'hown'))

data_frame = fetch_401k(return_type='data.frame')
# Construct DoubleMLData object from data.frame
dml_data_df = double_ml_data_from_data_frame(data_frame,
                                             y_col='net_tfa',
                                             d_cols='e401',
                                             x_cols=c('age', 'inc',
                                                      'educ', 'fsize',
                                                      'marr', 'twoearn',
                                                      'db', 'pira',
                                                      'hown'))
```

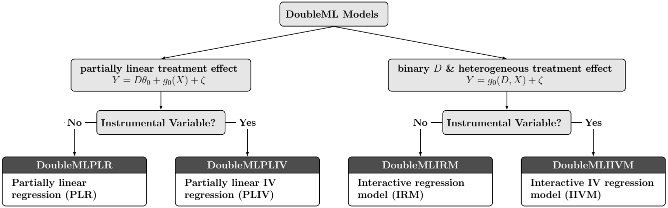
- **DoubleMLData** from matrix

```r
# Simulate data
set.seed(3141); n_obs = 500; n_vars = 100; theta = 3;
X = matrix(rnorm(n_obs*n_vars), nrow=n_obs, ncol=n_vars)
d = X[,1:3]%*%c(5,5,5) + rnorm(n_obs)
y = theta*d + X[, 1:3]%*%c(5,5,5) + rnorm(n_obs)

dml_data_sim =  double_ml_data_from_matrix(X, y, d)
```

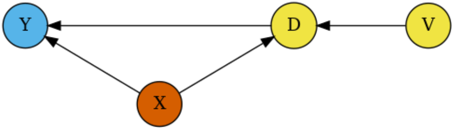
DoubleML Workflow: 2. Causal Model



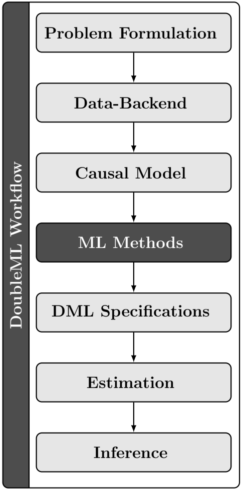
- Specify a **causal model**



- 401k data: Partially linear regression (PLR)



DoubleML Workflow: 3. ML Methods

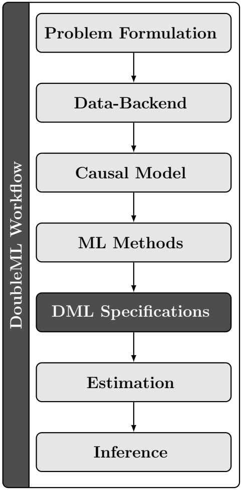


- Choose **ML methods** to approximate the nuisance functions
- PLR
  - $g_0(X) = \mathbb{E}(Y|X)$
  - $m_0(X) = \mathbb{E}(D|X)$
- Random forest from ranger/mlr3learners.

```
library(mlr3learners)
ml_g_rf = lrn("regr.ranger", max.depth = 7,
              mtry = 3, min.node.size = 3)
ml_m_rf = lrn("classif.ranger", max.depth = 5,
              mtry = 4, min.node.size = 7)
```

- Boosted trees from xgboost/mlr3extralearners

```
ml_g_xgb = lrn("regr.xgboost",
               objective = "reg:squarederror",
               eta = 0.1, nrounds = 35)
ml_m_xgb = lrn("classif.xgboost",
               objective = "binary:logistic",
               eval_metric = "logloss",
               eta = 0.1, nrounds = 34)
```



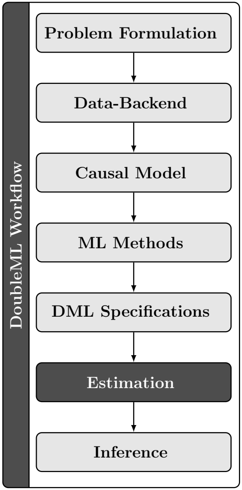
- **Initialize a DoubleMLPLR model**

```r
set.seed(123)
dml_plr_forest = DoubleMLPLR$new(dml_data,
                                 ml_g = ml_g_rf,
                                 ml_m = ml_m_rf,
                                 n_folds = 3)
```

- **Parametrize DoubleML models**
  - **Resampling** (repeated cross-fitting): Number of repetitions & folds
  - **DML algorithm**: dml1 vs. dml2
  - Neyman orthogonal **score function** (for PLR 'partialling out' or 'IV-type')

```r
set.seed(123)
dml_plr_forest = DoubleMLPLR$new(dml_data, ml_g = ml_g_rf,
                                 ml_m = ml_m_rf, n_folds=3, n_rep=1,
                                 score='partialling out',
                                 dml_procedure='dml2')
```

DoubleML Workflow: 5. Estimation



- **Estimation** of the DoubleML model

```
dml_plr_forest$fit()
```

- Estimated causal effect

```
dml_plr_forest$coef
    e401
8968.788
```

- Estimated standard error

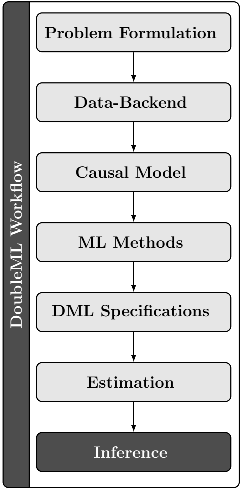
```
dml_plr_forest$se
    e401
1341.061
```

- **Summary** of the estimated effect

```
dml_plr_forest$summary()
[1] "Estimates␣and␣significance␣testing␣of␣the␣effect␣of␣target␣variables"
     Estimate. Std. Error t value Pr(>|t|)
e401      8969       1341   6.688 2.27e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '␣' 1
```

DoubleML Workflow: 6. Inference

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG



- Summary of the estimated effect

```
dml_plr_forest$summary()
[1] "Estimates and significance testing of the effect of target variables"
     Estimate. Std. Error t value Pr(>|t|)
e401      8969       1341   6.688 2.27e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- **Confidence interval(s)**

```
dml_plr_forest$confint()
        2.5 %   97.5 %
e401 6340.356 11597.22
```

- **Multiplier bootstrap**
  → relevant for multiple treatment effects!

```
dml_plr_forest$bootstrap()
```

- Confidence interval(s) based on the multiplier bootstrap

```
dml_plr_forest$confint(joint=TRUE)
        2.5 %   97.5 %
e401 6174.467 11763.11
```