Speaker Notes, Part 2

Philipp Bach

UseR! June 20, 2022

Speaker Notes

Speaker notes for second part of the tutorial *Causal Machine Learning with DoubleML* at the UseR!2022 Conference, June, 2022 by Philipp Bach (philipp.bach@uni-hamburg.de, University of Hamburg), Martin Spindler (martin.spindler@uni-hamburg.de, University of Hamburg), and Oliver Schacht (oliver.schacht@uni-hamburg.de).

Slide 1: Title Slide: Causal Machine Learning with DoubleML

- Welcome to the second part of our tutorial on Causal Machine Learning with DoubleML.
- This part: Introduction to the R package DoubleML

Slide 2: Title Slide: Introduction to DoubleML

- The slide shows the DoubleML logo - a double-headed rhino

Slide 3: Building Principles

- Let us start with the building principles for `DoubleML`. The three key ingredients to the DML framework also are also reflected by the implementation of our R package.
- We saw that a Neyman orthogonal score function plays a very important role in the theoretical framework. This is also reflected in our implementation. So the orthogonal score has a central role in the object-oriented implementation using R6 classes.
- We also want to provide the users access to high-quality ML methods. This is achieved by having a flexible interface to basically all ML learners available in the mlr3 ecosystem.
- We also have to include sample-splitting and do so by relying, again, on `mlr3`.

Slide 4: Dependencies and Installation

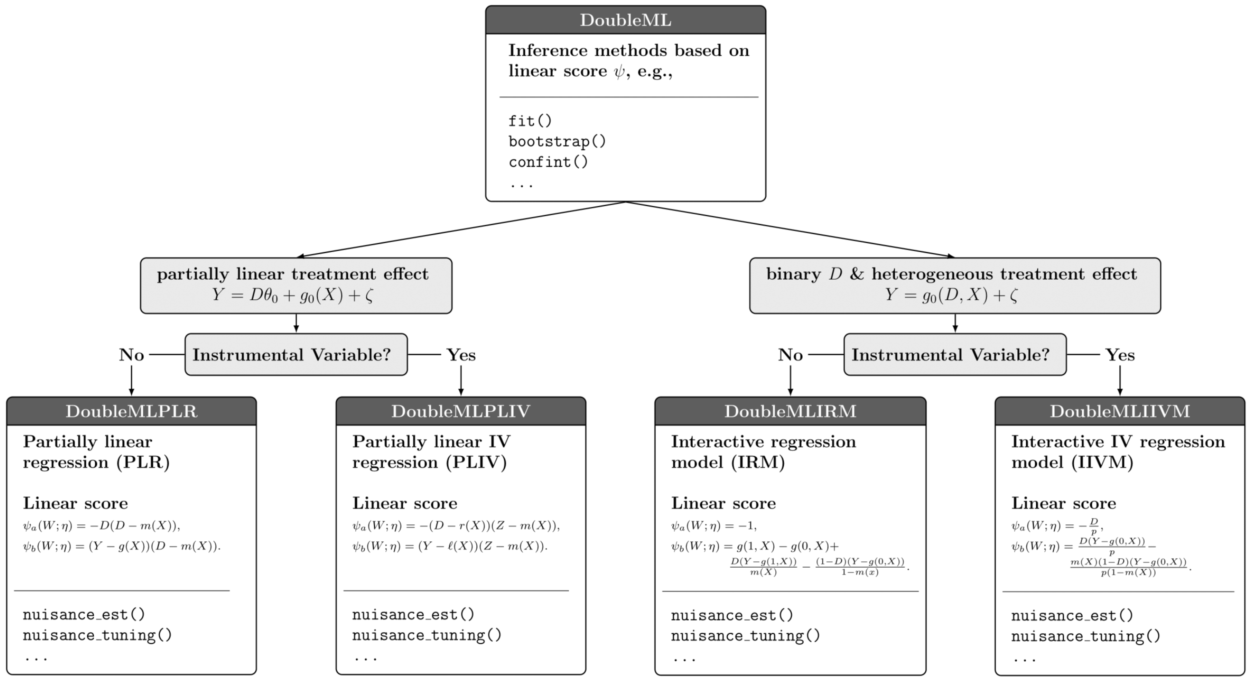
- On this slide, we list the main dependencies of our package. For tuning and fitting of the machine learning models, we build on the `mlr3`, the `mlr3learners`, and the `mlr3tuning` package. The object-orientation is based on `R6`. For efficient data handling, we use `data.table`.
- Installation from CRAN or from GitHUB can be done by the commands `install.packages("DoubleML")` and `remotes::install_github("DoubleML/doubleml-for-r")`

Slide 5: Why an Object-Oriented Implementation?

- Let's talk about the class structure and the causal models in `DoubleML`. As we have seen before, the orthogonal score function has a prominent role in the DML framework. If we have defined this score function for a model we are interested in, we can implement a lot of things in a very general rate, for example the estimation of parameters, computation of the score function $\psi()$, standard errors, confidence intervals and a multiplier bootstrap procedure.
- This is then implemented in an abstract base class called `DoubleML`. From this class, we can then inherit all the model classes. The only model specific parts are then how this score function is implemented.

Slide 6: Class Structure and Causal Models

- Let's talk about the model classes we have in `DoubleML`. We already saw the partially linear regression model before, but we also have other models in the package. For example, if we want to include a instrumental variable, we can use the partially linear instrumental variable regression model. Or if you are interested in modeling heterogeneous treatment effects, we also have the interactive regression model and its IV extension to estimate local average treatment effects.
- These model classes then inherit from the base class `DoubleML` and we can easily add new model classes in a similar way.



DoubleML
Inference methods based on linear score ψ, e.g.,
fit()
bootstrap()
confint()
...
partially linear treatment effect $Y = D\theta_0 + g_0(X) + \zeta$
binary D & heterogeneous treatment effect $Y = g_0(D, X) + \zeta$
No
Instrumental Variable?
Yes
No
Instrumental Variable?
Yes
DoubleMLPLR
Partially linear regression (PLR)
Linear score
$\psi_a(W;\eta) = -D(D - m(X)),$
$\psi_b(W;\eta) = (Y - g(X))(D - m(X)).$
nuisance_est()
nuisance_tuning()
...
DoubleMLPLIV
Partially linear IV regression (PLIV)
Linear score
$\psi_a(W;\eta) = -(D - r(X))(Z - m(X)),$
$\psi_b(W;\eta) = (Y - \ell(X))(Z - m(X)).$
nuisance_est()
nuisance_tuning()
...
DoubleMLIRM
Interactive regression model (IRM)
Linear score
$\psi_a(W;\eta) = -1,$
$\psi_b(W;\eta) = g(1,X) - g(0,X) + \frac{D(Y-g(1,X))}{m(X)} - \frac{(1-D)(Y-g(0,X))}{1-m(x)}.$
nuisance_est()
nuisance_tuning()
...
DoubleMLIIVM
Interactive IV regression model (IIVM)
Linear score
$\psi_a(W;\eta) = -\frac{D}{p},$
$\psi_b(W;\eta) = \frac{D(Y-g(0,X))}{p} - \frac{m(X)(1-D)(Y-g(0,X))}{p(1-m(X))}.$
nuisance_est()
nuisance_tuning()
...

Figure 1: Description of Figure (alt-text): The figure shows a diagramm of the object class structure in `DoubleML`. On top, there is a box illustrating the base class `DoubleML`. In this class, inference methods that are based on a linear score function ψ are implemented, for example the methods `fit()`, `bootstrap()` and `confint()` for parameter estimation and construction of confidence intervals. Below, there are four different boxes indiciating four different model classes: `DoubleMLPLR` for partially linear regression models, `DoubleMLPLIV` for partially linear instrumental variable regression, `DoubleMLIRM` for an interactive or non-parametric regression model and `DoubleMLIIVM` for an IV-version of this interactive regression model. Each of these models is characterized by a different score function ψ.

Slide 7: Advantages of the Object-Orientation

- Let's talk about the advantages of this object-oriented implementation.
- First of all, it gives the user a very high flexibility with regards to the model specifications. For example, you basically choose from a great variety of ML learers to estimate the nuisance functions, you can easily alter the resampling scheme, for example in terms of the number of folds used in cross-fitting. You can also choose between different DML algorithms and different Neyman orthogonal scores.
- A second key advantage is that the package is easily extendible. Users can easily add new causal model classes that inherit from the base class `DoubleML`. You can also add other resampling schemes and score functions.

Slide 8: Title Slide: Getting Started with DoubleML!

Slide 9: Installation

- Now, let's get started with our first steps in DoubleML
- You can install the latest CRAN release of DoubleML from CRAN via `install.packages("DoubleML")`
- Alternatively, you can also install the development version from GitHub by typing `remotes::install_github("DoubleM`

Slide 10: Data Example - Demand Estimation

- The slide presents an illustration of demand estimation

- In our introduction to the DoubleML package, we'll follow a real-data example on estimation of price Elasticity of demand.
- The causal problem we will consider is as follows: We set up a regression model that allows us to estimate the price elasticity of demand. This quantity tells us by how much the demanded quantity of a product changes due to a change in the product price.
- Identification of the elasticity is based on an selection-on-observables approach, i.e., we flexibly include product characteristics in our regression model in order to account for confounding.
- Causal diagram

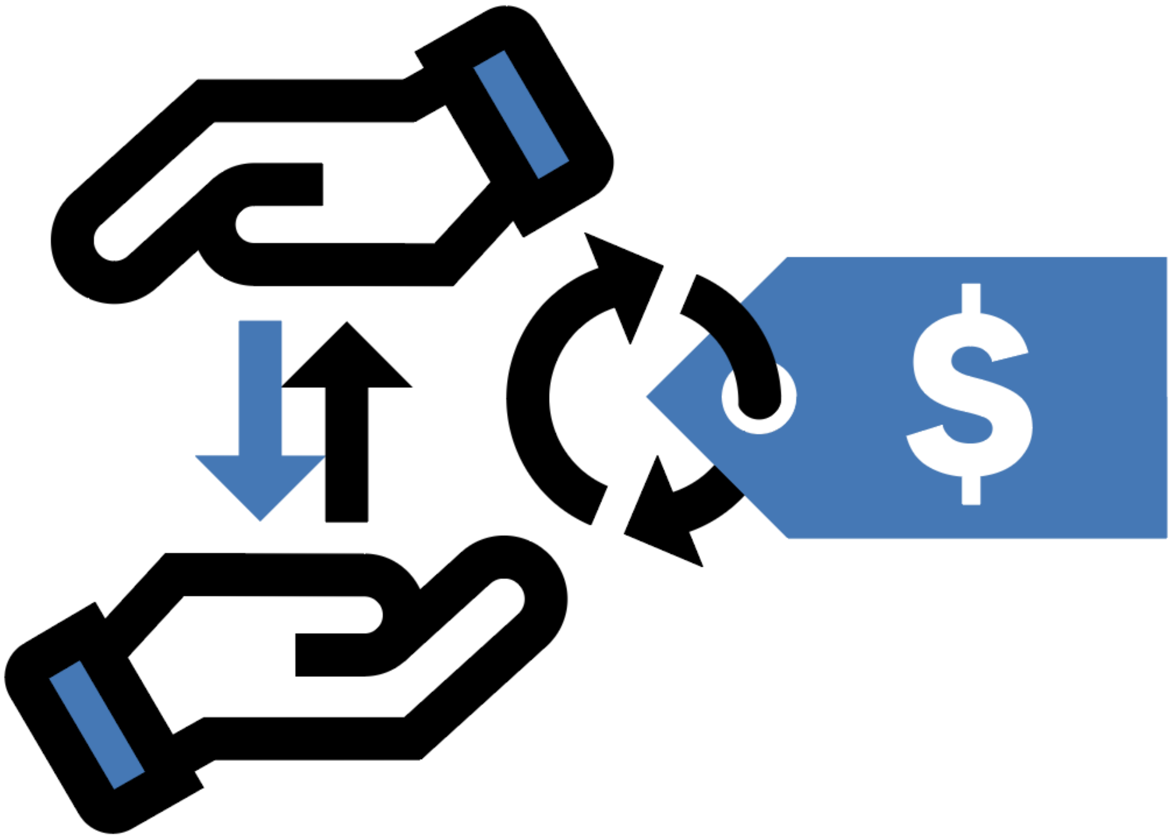


Figure 2: An illustration of demand estimation. On the left hand side two hands are displayed on top of each other. Between the hands there are two arrows showing up and down. On the right hand side, there is a price tag with a dollar sign attached to a circular graph.

$$\begin{tikzcd}
 & X \arrow[d] \arrow[dr] & \\
U \arrow[r] & \text{dLnP} \arrow[r] & \text{dLnQ}
\end{tikzcd}$$

- We use a real data set of an online retailer that has been published on **kaggle**](https://www.kaggle.com/vijayuv/onlineretail). A **preprocessing notebook is available online**
- The notebook we consider is based on **blogpost by Lars Roemheld (Roemheld, 2021)**

Slide 11: Title Slide: Hands On! Interactive Breakout Sessions

- The slide shows an illustration of AB testing

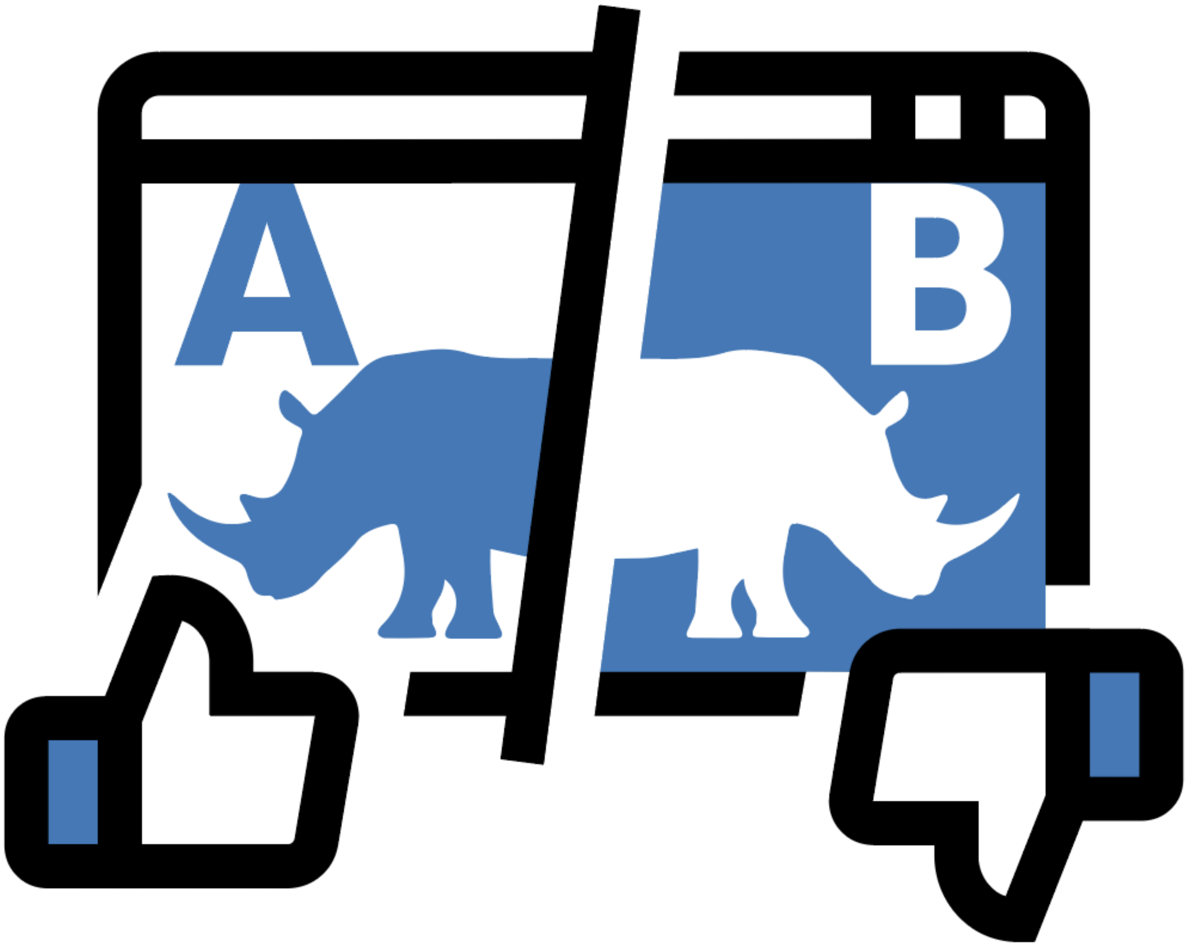


Figure 3: An illustration of AB testing. A stylized browser window shows a double-headed rhino which is a variant of the DoubleML package logo. The screen is divided vertically in two parts. The left part of the screen has the tag 'A' and differs from the right part called 'B' in that the colors are inverted.

-

Slide 12: AB Testing

- In the interactive session, you'll apply DoubleML in the context of AB testing.
- The causal problem considered is to assess the effect a new ad design has on the sales of a web shop, on average.
- In the example, we have to rely on an observational framework, i.e., the data set has been collected. Again, we will use a selection-on-observables approach, i.e., flexibly account for confounding variables, V

- We will use a data set from the 2019 Atlantic Causal Inference Challenge (ACIC). Although the data set has not explicitly been created to match an AB test setting, we'll re-frame it in this setting in order to have a tractable and realistic scenario.
- The causal diagram in this case is

$$\begin{tikzcd}
 & V \arrow[d] \arrow[dr] & \\
U \arrow[r] & A \arrow[r] & Y
\end{tikzcd}$$

Slide 13: Online Resources

- When you solve the case study, you might find the following resources helpful:
- The notebook is organized according to the **DoubleML Workflow**
- Extensive **User Guide** available via **docs.doubleml.org**
- **Documentation for the R Package DoubleML** available via **docs.doubleml.org/r/stable/**
- R vignette, Bach et al. (2021) available via **arxiv**

Slide 14: Quickstart to R6

- If you don't have experience with object orientation in R, we'll provide some notes on a quickstart in R6
- A short introduction to the `R6` packages is **available here**.
- To create a new instance of a class, you can call the `$new()` method. Here, we provide an example on how to initiate a data backend for `DoubleML`, i.e., an new instance of the class `DoubleMLData`

```r
# Example create a backend (class DoubleMLData)
library(DoubleML)
df = make_plr_CCDDHNR2018(return_type = "data.table")
obj_dml_data = DoubleMLData$new(df,
                                y_col = "y",
                                d_cols = "d")
```

Slide 15: Quickstart to R6

- Once you created an object, you can call methods or access fields, for example the number of observations of the data backend

```r
obj_dml_data$n_obs
```

```
## [1] 500
```

or to call the `$print()` method defined for this object class

```r
obj_dml_data$print()
```

```
## ================= DoubleMLData Object ==================
##
##
## ------------------ Data summary      ------------------
## Outcome variable: y
## Treatment variable(s): d
## Covariates: X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14, X15, X16, X17, X18, X19, X2
## Instrument(s):
## No. Observations: 500
```

Slide 15: Quickstart to R6

- Debugging with R6 is also a bit different than in functional R
- A guide on how to debug with R6 is **avaialable online**
- Enabling debugging for all future instances of a class

```r
DoubleMLData$debug("initialize")
obj_dml_data = DoubleMLData$new(df,
                                y_col = "y",
                                d_cols = "d")
```

- Debugging methods in individual objects

```r
debug(obj_dml_data$print)
obj_dml_data$print()
```

Slide 17: Quickstart - Creating learners in mlr3

- We briefly list the commands required for generating learners in mlr3
- Install and load `mlr3` package

```
install.packages("mlr3")
library(mlr3)
```

- Create a learner

```
lm_learner = LearnerRegrLM$new()
```

```
lm_learner = lrn("regr.lm")
lm_learner
```

```
## <LearnerRegrLM:regr.lm>
## * Model: -
## * Parameters: list()
## * Packages: mlr3, mlr3learners, stats
## * Predict Type: response
## * Feature types: logical, integer, numeric, factor, character
## * Properties: loglik, weights
```