Appendix: DoubleML の活用

川田恵介

2025-08-08

1 DoubleML

1.1 概要

- 本講義で紹介した Residuals regression や AIPW を実装する包括的なパッケージ
  - ddml などに比べ、多機能
  - 利用者が多く、信頼性が高い
  - R/Python で提供されている (Homepage)
  - 機械学習の代表的なパッケージ (mlr3 / scikit-learn) をベースとし、学ぶ”コスパ”が良い

1.2 習熟が必要な点

- R では比較的少数のパッケージで採用される“オブジェクト指向プログラミング（Object-Oriented Programming）”に慣れる必要がある
  - ざっくり、データと”データを操作する関数”が同じ object にまとめられている

2 mlr3

2.1 SetUp

```
set.seed(111)
library(tidyverse)
library(mlr3verse)

data <-read_csv("Public/Data.csv")
```

2.2 推定方法(learner)の定義

```
OLS <- lrn("regr.lm") # OLS
RF <- lrn("regr.ranger") # Random Forest
```

2.3 推定問題(task)の定義

```
task <- as_task_regr(
  x = data, # 用いるデータ
  target = "Y" # Y
  )
```

2.4 推定

```
group <- partition(
  task,
  ratio = 0.8
  ) # サンプル分割

OLS$train(
  task,
  group$train # 訓練データのみ使用し、OLS推定
  )
```

- イメージ: “OLS object”から呼び出した”train” method (推定方法,OLS) を、task に適用し、その推定結果を”OLS object”に保存

2.5 予測

```
pred_OLS <- OLS$predict(
  task,
  group$test # テストデータに、予測を適用
  )

pred_OLS
```

```
── <PredictionRegr> for 2262 observations:
──────────────────────────────────────
 row_ids    truth response
       1 18.35881 17.90616
       2 18.42068 18.18142
       4 18.40048 18.10534
     ---      ---      ---
   11305 17.24950 17.56632
   11310 17.18281 17.11940
   11311 17.18281 17.26616
```

- イメージ: object から呼び出した”predict” method (予測)を task に適用

2.6 評価

```
pred_OLS$score() # 平均二乗誤差
```

```
regr.mse
0.104394
```

2.7 method chaining

- Random Forest を用いて予測モデルを推定し、予測値を算出し、評価する

```
RF$train(task,group$train)$predict(task, group$test)$score()
```

```
  regr.mse
0.09227832
```

2.8 利用可能な推定方法

- コアな手法は、mlr3learners に収録
  - 大きく回帰問題に対応する手法 (regr.)と分類問題に対応する手法 (classif.) に分けられている
- 試験的な手法は、mlr3extralearners に収録

3 Double ML

3.1 SetUp

```
library(tidyverse)
library(DoubleML)
library(mlr3verse)

data <- read_csv("Public/Data.csv")
```

3.2 Task

```
task <- double_ml_data_from_matrix(
  X = data |> select(Size,Tenure,Distance), # X
  y = data$Y, # Y
  d = data$Reform # D
)
```

3.3 推定方法

```
PLR <- DoubleMLPLR$new( # 残差回帰
  task,
```

```
  ml_l = lrn("regr.lm"), # Yの推定方法
  ml_m = lrn("regr.lm"), # Dの推定方法
  n_folds = 2 # データの分割数
  )

AIPW <- DoubleMLIRM$new( # AIPW
  task,
  ml_g = lrn("regr.lm"), # Yの推定方法
  ml_m = lrn("classif.log_reg"), # Dの推定方法 (Logit)
  n_folds = 2
  )
```

3.4 推定: PLR

```
PLR$fit()

PLR
```

```
================== DoubleMLPLR Object ==================


------------------ Data summary      ------------------
Outcome variable: y
Treatment variable(s): d
Covariates: X1, X2, X3
Instrument(s):
Selection variable:
No. Observations: 11311

------------------ Score & algorithm ------------------
Score function: partialling out
DML algorithm: dml2

------------------ Machine learner   ------------------
ml_l: regr.lm
ml_m: regr.lm

------------------ Resampling        ------------------
No. folds: 2
No. repeated sample splits: 1
Apply cross-fitting: TRUE

------------------ Fit summary       ------------------
 Estimates and significance testing of the effect of target variables
  Estimate. Std. Error t value Pr(>|t|)
d  0.174548   0.008989   19.42   <2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.5 推定: AIPW

```
AIPW$fit()

AIPW
```

```
================== DoubleMLIRM Object ==================


------------------ Data summary      ------------------
Outcome variable: y
Treatment variable(s): d
Covariates: X1, X2, X3
Instrument(s):
Selection variable:
No. Observations: 11311

------------------ Score & algorithm ------------------
Score function: ATE
DML algorithm: dml2

------------------ Machine learner   ------------------
ml_g: regr.lm
ml_m: classif.log_reg

------------------ Resampling        ------------------
No. folds: 2
No. repeated sample splits: 1
Apply cross-fitting: TRUE

------------------ Fit summary       ------------------
 Estimates and significance testing of the effect of target variables
  Estimate. Std. Error t value Pr(>|t|)
d  0.127400   0.008197   15.54   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.6 BLP の推定

- $E[Y \mid D = 1, X] - E[Y \mid D = 0, X]$ についての線型モデルも容易に推定できる

```
psu_Y <- AIPW$psi_b
```

```
estimatr::lm_robust(psu_Y ~ Size + Tenure, data)
```

```
                Estimate   Std. Error    t value     Pr(>|t|)      CI Lower
(Intercept) -0.0702202684 0.0197151798 -3.5617361 3.699251e-04 -0.1088654472
Size         0.0001625683 0.0003780547  0.4300128 6.671945e-01 -0.0005784845
Tenure       0.0093495224 0.0006410844 14.5839190 9.635232e-48  0.0080928856
                CI Upper    DF
(Intercept) -0.0315750896 11308
Size         0.0009036212 11308
Tenure       0.0106061592 11308
```

3.7 発展: 繰り返し分割の活用

- “サンプル分割の結果”は、推定結果に影響を与えうる
  - 漸近的には無視できるが、有限サンプルにおいては無視できない場合も多い
- 対応: “サンプル分割 → 推定”を何度か繰り返し、推定値の中央値と対応する標準誤差を報告する
  - 計算時間が増えることに注意

3.8 実装

```
PLR_robust <- DoubleMLPLR$new( # 残差回帰
  task,
  ml_l = lrn("regr.lm"),
  ml_m = lrn("regr.lm"),
  n_folds = 2,
  n_rep = 10 # 10回繰り返す
  )

PLR_robust$fit()
```

4 pipeline の活用

4.1 pipe

- mlr3pipeline を活用すると複雑な推定手法を定義できる
  - stacking
  - データの前処理 → 推定

4.2 SetUp

```
library(tidyverse)
library(DoubleML)
```

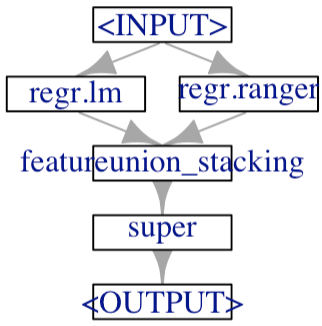
```
library(mlr3verse)
library(mlr3pipelines)

data <- read_csv("Public/Data.csv")
```

4.3 stacking

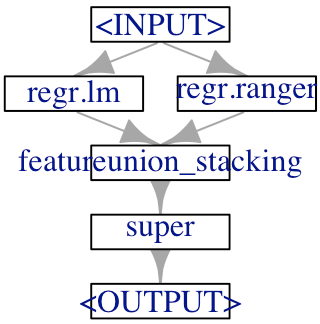
```
stacking <- pipeline_stacking(
  base_learners = list(lrn("regr.lm"),lrn("regr.ranger")), # 個別の推定方法
  super_learner = lrn("regr.lm", id = "super"), # 推定値の集計方法
  folds = 3, # 分割数
  use_features = FALSE # 元の変数は利用しない
  )

stacking$plot()
```



4.4 確認

```
stacking$plot()
```



4.5 PLR

```
learn_stack <- as_learner(stacking) # learner化

PLR_stack <- DoubleMLPLR$new( # 残差回帰
  task,
  ml_l = learn_stack$clone(),
  ml_m = learn_stack$clone(),
  n_folds = 2,
  n_rep = 1
  )

PLR_stack$fit()
```

4.6 VS ddml

- 講義では short stacking (Ahrens et al., 2025) を用いた方法を紹介し、ddml で実装
- 川田の知る限り、doubleml で short stacking の実装は難しい
  - ただし doubleml の実装は、計算時間の問題を除けば、より一般的に利用できる
    - 交差推定を利用する stacking をさらに交差推定している

Bibliography

Ahrens, A. et al. (2025) “Model averaging and double machine learning,” Journal of Applied Econometrics, 40(3), pp. 249–269.