StatML.io CDT: Causality Module

Robin J. Evans

Imperial College London and University of Oxford
March 2024

Outline

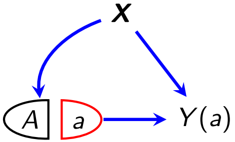
1. Machine Learning Methods

Post Double Selection Inference

1. Machine Learning Methods
   - Post Double Selection Inference
   - Double Machine Learning

Post 'Double Selection' Inference

Suppose we have the following set up, where $\boldsymbol{X}$, is high-dimensional (say $|\boldsymbol{X}| = p$).



It is clear that we can **identify** the causal effect of A on Y, since assuming independent observations and the model implied by the SWIG:

$$\mathbb{E}Y(a) \ = \ \sum_{\boldsymbol{x}} P(\boldsymbol{x}) \cdot \mathbb{E}[Y \,|\, a, \boldsymbol{x}] \ = \ \mathbb{E}\left[\frac{Y \mathbb{1}_{\{A=a\}}}{P(A = a \,|\, \boldsymbol{X})}\right];$$

however, statistically we may still have difficulties.

- We do not know what form the expressions for $\mathbb{E}[Y \,|\, a, \boldsymbol{x}]$, $P(\boldsymbol{x})$, or $P(a \,|\, \boldsymbol{x})$ should take.
- Even if we knew the families, actually estimating the parameters may be infeasible with a finite dataset of reasonable size.

Frisch-Waugh-Lovell Theorem

Suppose we have n i.i.d. observations $(\boldsymbol{X}_i, A_i, Y_i)$ such that

$$A_i = \alpha^T \boldsymbol{X}_i + \delta_i \qquad\qquad Y_i = \beta A_i + \gamma^T \boldsymbol{X}_i + \varepsilon_i,$$

where $\boldsymbol{X}_i$ has fewer than $n - 1$ entries.

Consider two different ways of obtaining an estimate of β:

1. regress Y on $\boldsymbol{X}$ and A using OLS, and look at $\hat{\beta}$;
2. regress Y on $\boldsymbol{X}$ to obtain residual r_Y; and then A on $\boldsymbol{X}$ to obtain r_A; then regress r_Y on r_A, and take the linear coefficient $\tilde{\beta}$.

Theorem (Frisch and Waugh (1933), Lovell (1963))

The estimates for β from methods 1 and 2 are the same.

Intuition

Why does this result hold?

Proof.

Note that $r_A = A - \hat{\alpha}^T \boldsymbol{X}$, so $r_A \perp\!\!\!\perp \boldsymbol{X}$.
Then

$$\begin{aligned} \mathbb{E}[Y \mid \boldsymbol{X}, A] &= \beta A + \gamma^T \boldsymbol{X} \\ &= \beta(r_A + \alpha^T \boldsymbol{X}) + \gamma^T \boldsymbol{X} \\ &= \beta r_A + (\alpha + \gamma)^T \boldsymbol{X}. \end{aligned}$$

Then, since $\boldsymbol{X} \perp\!\!\!\perp r_A$, we must have that regressing Y on $\boldsymbol{X}$ gives an estimate of $\alpha + \gamma$.
Hence

$$\mathbb{E} r_Y = \beta \mathbb{E} r_A,$$

giving the result. $\square$

Sparsity

Suppose that we have

$$\mathbb{E}[A \mid \boldsymbol{X} = \boldsymbol{x}] = \alpha^T \boldsymbol{x}$$
$$\mathbb{E}[Y \mid A = a, \boldsymbol{X} = \boldsymbol{x}] = \beta a + \gamma^T \boldsymbol{x}.$$

Assume also that $\log p = o(n^{1/3})$ and there exist subsets $\boldsymbol{B}$ and $\boldsymbol{D}$ of size at most $s_n \ll n$ such that:

$$\mathbb{E}[A \mid \boldsymbol{x}] = \alpha_{\boldsymbol{B}}^T \boldsymbol{x} + r_n$$
$$\mathbb{E}[Y \mid A = a, \boldsymbol{X} = \boldsymbol{x}] = \beta a + \gamma_{\boldsymbol{D}}^T \boldsymbol{x} + t_n,$$

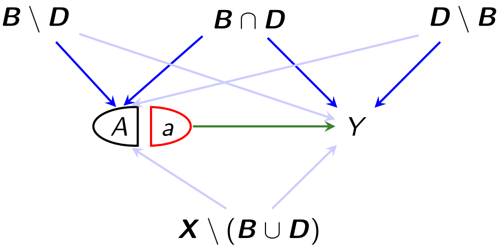
where the approximation error is stochastically smaller than the estimation error: i.e.

$$\mathbb{E}\|r_n\|_2 \lesssim \sqrt{\frac{s_n}{n}} \qquad \text{and} \qquad \mathbb{E}\|t_n\|_2 \lesssim \sqrt{\frac{s_n}{n}}.$$

In other words, a much smaller subset of covariates is sufficient to **approximately** make A and Y unconfounded.

Post 'Double Selection' Inference

Graphical representation:



The idea is that if we account for variables in **both $\boldsymbol{B}$ and $\boldsymbol{D}$**, then we will be guaranteed to have good control of the bias in estimating β.

In principle we can use any consistent selection method to choose $\boldsymbol{B}$ and $\boldsymbol{D}$. In practice, Belloni et al. recommend a version of the lasso.

Post 'Double Selection' Inference

Here we perform a simulated example. Suppose that

$$A_i = \alpha \sum_{i=1}^{7} X_i + \delta_i$$

$$Y_i = \beta A_i + \gamma \sum_{i=4}^{10} X_i + \varepsilon_i$$

where $\delta_i, \varepsilon_i \overset{\text{i.i.d.}}{\sim} N(0,1)$ (independently), and we are given 1000 covariates in $\boldsymbol{X}$, where each $X_{ij} \sim N(0,1)$ independently.

Set $\beta = \gamma = 2$ and $\alpha = 1$, and pick $n = 100$.

Post ‘Double Selection’ Inference

```
alpha <- 1
gamma <- beta <- 2
n <- 100; p <- 1000

## simulate data
set.seed(123)
Z <- matrix(rnorm(n*p), n, p)
X <- Z %*% c(rep(alpha, 7), rep(0,p-7)) + rnorm(n)
Y <- Z %*% c(rep(0,3), rep(gamma, 7), rep(0,p-10)) + beta*X + rnorm(n)
dat <- data.frame(Y=Y, X=X, Z)
names(dat) <- c("Y","X",paste0("Z",seq_len(p)))

head(dat[,1:9])
```

```
        Y      X      Z1     Z2     Z3     Z4      Z5      Z6      Z7
1  -1.932  0.876 -0.5605 -0.710  2.199 -0.715 -0.0736 -0.6019  1.0740
2 -11.460  0.227 -0.2302  0.257  1.312 -0.753 -1.1687 -0.9937 -0.0273
3   0.821  0.408  1.5587 -0.247 -0.265 -0.939 -0.6347  1.0268 -0.0333
4  -0.752 -1.633  0.0705 -0.348  0.543 -1.053 -0.0288  0.7511 -1.5161
5  -4.478 -1.284  0.1293 -0.952 -0.414 -0.437  0.6707 -1.5092  0.7904
6  -2.355  0.906  1.7151 -0.045 -0.476  0.331 -1.6505 -0.0951 -0.2107
```

Post 'Double Selection' Inference

We can try a naïve model, and obtain the wrong answer.

```
sum_lm <- summary(lm(Y ~ X, data=dat))
sum_lm$coef


            Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.244      0.492   0.496 6.21e-01
X              3.067      0.184  16.649 2.52e-30


coef <- sum_lm$coef
```

Notice that the estimate $\hat{\beta} = 3.07$ is not within 2 s.e.s (0.37) of $\beta = 2$.

Post ‘Double Selection’ Inference

Then we can try using the R package `hdm`, which implements double selection.

```
library(hdm) ## library for implementation
lasso_out = rlassoEffect(y=dat[,"Y",drop=FALSE],
                         d=dat[,"X",drop=FALSE],
                         x=Z, method="double selection")

sum_out <- summary(lasso_out)
sum_out


[1] "Estimates and significance testing of the effect of target variables"
  Estimate. Std. Error t value Pr(>|t|)
X     2.018      0.119    16.9   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note this solution $\tilde{\beta} = 2.02$, is (well) within two s.e.s (0.24) of $\beta = 2$.

Post 'Double Selection' Inference: Application

Let us try applying double selection to a wage dataset.

```
X <- model.matrix(~ -1 + female + (widowed +divorced + separated +
                  nevermarried + hsd08 + hsd911 + hsg + cg + ad + mw + so +
                  we + exp1 + exp2 + exp3)^2, data = cps2012)
X <- X[, apply(X, 2, var) != 0] # exclude all constant variables
y <- cps2012$lnw
effects_female <- rlassoEffects(x = X, y = y, index = "female")
summary(effects_female)
```

```
[1] "Estimates and significance testing of the effect of target variables"
       Estimate. Std. Error t value Pr(>|t|)
female  -0.28067    0.00692   -40.5   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Post 'Double Selection' Inference: Application

Now let's try fitting the other covariates too (note some are causally subsequent to sex).

```
data(cps2012)
X <- model.matrix(~ -1 + female + female:(widowed + divorced + separated +
                  nevermarried +hsd08 + hsd911 + hsg + cg + ad + mw + so +
                  we + exp1 + exp2 + exp3) + (widowed +divorced + separated +
                  nevermarried + hsd08 + hsd911 + hsg + cg + ad + mw + so +
                  we + exp1 + exp2 + exp3)^2, data = cps2012)
X <- X[, apply(X, 2, var) != 0] # exclude all constant variables
index.gender <- grep("female", colnames(X))
y <- cps2012$lnw
```

Post ‘Double Selection’ Inference: Application

```
effects_female <- rlassoEffects(x = X, y = y, index = index.gender)
summary(effects_female)


[1] "Estimates and significance testing of the effect of target variables"
                     Estimate. Std. Error t value Pr(>|t|)
female               -0.15492    0.05016   -3.09  0.00201 **
female:widowed        0.13610    0.09066    1.50  0.13332
female:divorced       0.13694    0.02218    6.17  6.7e-10 ***
female:separated      0.02330    0.05321    0.44  0.66144
female:nevermarried   0.18685    0.01994    9.37  < 2e-16 ***
female:hsd08          0.02781    0.12091    0.23  0.81809
female:hsd911        -0.11934    0.05188   -2.30  0.02144 *
female:hsg           -0.01289    0.01922   -0.67  0.50252
female:cg             0.01014    0.01833    0.55  0.58011
female:ad            -0.03046    0.02181   -1.40  0.16241
female:mw            -0.00106    0.01919   -0.06  0.95581
female:so            -0.00818    0.01936   -0.42  0.67247
female:we            -0.00423    0.02117   -0.20  0.84176
female:exp1           0.00494    0.00780    0.63  0.52714
female:exp2          -0.15952    0.04530   -3.52  0.00043 ***
female:exp3           0.03845    0.00786    4.89  1.0e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

References

Belloni, A., Chernozhukov, V. and Hansen, C. (2014). Inference on treatment effects after selection among high-dimensional controls. *The Review of Economic Studies*, 81(2), 608–650.

Frisch, R. and F.V. Waugh (1933). Partial time regression as compared with individual trends. *Econometrica* 1 (October): 387–401.

Lovell, M.C. (1963). Seasonal adjustment of economic time series and multiple regression analysis. *JASA* 58 (December): 993–1010.

Double Machine Learning

1. Machine Learning Methods
   - Post Double Selection Inference
   - Double Machine Learning

Double Machine Learning

Double (or **debiased**) **machine learning** is an increasingly common approach to estimating causal effects. See, e.g. Chernozhukov et al. (2018).

The basic idea is the same as the approach of Belloni et al. (2014).

We estimate separate **high-dimensional models** for the treatment and outcome.

The methods make extensive use of **cross-fitting**, i.e. splitting the data into separate components and using each to predict the other.

This allows for estimation while preventing **over-fitting**.

Mathematically speaking, much more **complicated models** can be used but still give an unbiased estimator of a (low-dimensional) causal effect.

Conditions for Double ML

A crucial condition for double ML to work is **Neyman orthogonality**, which says that the derivative of the estimating equation (at the true parameters) with respect to any nuisance parameters should be zero.

Suppose our score function is $\psi(W; \theta, \eta)$, with parameters of interest θ and nuisance parameters η. Then we need:

$$\frac{\partial}{\partial \eta} \mathbb{E}\psi(W; \theta_0, \eta)\Big|_{\eta=\eta_0} = 0,$$

where (θ_0, η_0) are the true parameters.

If we are given a score function that is **not** Neyman orthogonal, we can often change it to become so.

Conditions for Double ML

Consider the linear model example, where the usual score is

$$\tilde{\psi}_\beta(W;\beta,\gamma) = (Y - \beta A - \gamma^T \boldsymbol{X}) \cdot A$$
$$\tilde{\psi}_\gamma(W;\beta,\gamma) = (Y - \beta A - \gamma^T \boldsymbol{X}) \cdot \boldsymbol{X}.$$

Suppose we consider a directional derivative $\delta \cdot h$ with $h \in \mathbb{R}^{|\boldsymbol{X}|}$, then we have

$$\begin{aligned}
&\frac{\partial}{\partial\gamma}\tilde{\psi}_\beta(W;\beta,\gamma_0 + \delta h)\Big|_{\delta\to 0} \\
&= \lim_{\delta\to 0} \frac{(Y - \beta A - (\gamma_0 + \delta h)^T \boldsymbol{X}) \cdot A - (Y - \beta A - \gamma_0^T \boldsymbol{X}) \cdot A}{\delta} \\
&= -h^T \boldsymbol{X}.
\end{aligned}$$

In particular, this is **not** zero!

Conditions for Double ML

Now, we can reparametrize the nuisance parameter γ as $\eta = (\gamma, \mu)$, where we choose μ so that the new score for β is

$$\begin{aligned}\psi_\beta(W; \beta, \eta) &= \tilde{\psi}_\beta(W; \beta, \gamma) - \mu^T \tilde{\psi}_\gamma(W; \beta, \gamma) \\ &= (Y - \beta A - \gamma^T \boldsymbol{X})(A - \mu^T \boldsymbol{X}).\end{aligned}$$

If we pick $\mu = \alpha$, then note that the expectation of second factor is 0!

Hence, **small** errors in the estimation of γ and α will **not** affect the estimate of β.

In particular:

$$\frac{\partial}{\partial \gamma} \psi_\beta(W; \beta, \gamma, \alpha) = -\boldsymbol{X}(A - \alpha^T \boldsymbol{X})$$

$$\text{and} \quad \frac{\partial}{\partial \alpha} \psi_\beta(W; \beta, \gamma, \alpha) = -\boldsymbol{X}(Y - \beta A - \gamma^T \boldsymbol{X}),$$

and these both have expectation 0.

Moral: Neyman orthogonality is very helpful for robustness to misspecification.

401(k) Example

Chernozhukov et al. (2018) analyse data on 401(k) savings plans, and whether eligibility to enroll leads to an increase in net assets.

They consider a dataset of 9,915 individuals, measuring:

- **age** age in years;
- **inc** income;
- **educ** years of education;
- **fsize** family size;
- **marr** indicator of being married;
- **twoearn** two earners in household;
- **db** member of defined benefit pension scheme;
- **pira** eligible for Individual Retirement Allowance;
- **hown** homeowner.

DML for 401(k) Example

```
library(DoubleML)
library(mlr3)
library(data.table)
library(dplyr)

## note that the DoubleML package uses data.table objects
dat <- fetch_401k(return_type = "data.table", instrument = TRUE)

# Initialize DoubleMLData (data-backend of DoubleML)
dml = DoubleMLData$new(dat,
                       y_col = "net_tfa",
                       d_cols = "e401",
                       x_cols = c("age", "inc", "educ", "fsize",
                        "marr", "twoearn", "db", "pira", "hown"))
mod <- DoubleMLIRM$new(dml,
              ml_m = lrn("classif.cv_glmnet", s = "lambda.min"),
              ml_g = lrn("regr.cv_glmnet",s = "lambda.min"),
              n_folds = 10, n_rep = 10)
mod$fit()   ## fit the model
```

```
c(beta=mod$coef, se=mod$se)
```

```
beta.e401   se.e401
     1669      3752
```

DML for 401(k) Example

We can also try using a more flexible set of covariates.

```
## add quadratic terms to age, income, education and family size
formula_flex = formula(" ~ -1 + poly(age, 2, raw=TRUE) +
  poly(inc, 2, raw=TRUE) + poly(educ, 2, raw=TRUE) +
  poly(fsize, 2, raw=TRUE) + marr + twoearn + db + pira + hown")
features_flex = data.frame(model.matrix(formula_flex, dat))
model_data = data.table("net_tfa" = dat[, net_tfa],
                        "e401" = dat[, e401], features_flex)

## initialize and fit model
dml_f <- DoubleMLData$new(model_data, y_col = "net_tfa",
                          d_cols = "e401")
mod_f <- DoubleMLIRM$new(dml_f,
            ml_m = lrn("classif.cv_glmnet", s = "lambda.min"),
            ml_g = lrn("regr.cv_glmnet",s = "lambda.min"),
            n_folds = 10, n_rep = 5)
mod_f$fit()
```

We obtain a much smaller standard error.

```
c(beta=mod_f$coef, se=mod_f$se)


beta.e401   se.e401
     8538      1258
```

References

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

References

Pearl, J. *Causality: Models, Reasoning, and Inference.* 3rd Ed. Cambridge, 2009.

Spirtes, P, Glymour, C, Scheines R. *Causation, Prediction, and Search.* Lecture Notes in Statistics 81, Springer-Verlag, 2000.

Wright, S. The theory of path coefficients. *Genetics*, 8: 239–255, 1923.

Wright, S. The method of path coefficients. *Annals of Mathematical Statistics*, 5(3): 161–215, 1934.