Macro effects

Ľuboš Perniš

09/04/2019

Macro effects

We start again by creating the initial and the target location.

```
source('functions/causalMatchFNN.R')
library(causalTree)
```

```
## Loading required package: rpart
```

```
## Loading required package: rpart.plot
```

```
## Loading required package: data.table
```

```
### First create data
# This helper function generates y1
generate_y1 <- function(treatment_function) {
  return(y0 + eval(parse(text  = treatment_function)))
}

treatment_function <- '2 + 10 * x1'

## Create the synth dataset for D = 0
# Create covariate x1
set.seed(123) # For reproducibility
uniform <- runif(n = 400, min = 0, max = 10)
set.seed(123)
normal <- rnorm(n = 100, mean = 8)
x1 <- c(uniform, normal)
x1 <- x1[x1 <= 10 & x1 >= 0]
# Create y0
y0 <- x1
# Define treatment
y1 <- generate_y1(treatment_function)

# Bind together to create simulation.1 d= 0 dataset
d0 <- as.data.frame(cbind(x1, y0, y1))

## Create the synth dataset for D = 1
# Create covariate x1
set.seed(1234) # For reproducibility
uniform <- runif(n = 400, min = 0, max = 10)
set.seed(1234)
normal <- rnorm(n = 100, mean = 2)
x1 <- c(uniform, normal)
x1 <- x1[x1 <= 10 & x1 >= 0]
# Create y0
y0 <- x1
# Define treatment
```

```
y1 <- generate_y1(treatment_function)

# Bind together to create simulation.1 d= 0 dataset
d1 <- as.data.frame(cbind(x1, y0, y1))

# Clean the workspace
rm(x1, y1, y0, uniform, treatment_function, normal, generate_y1)

### Assign treatment to D = 0
set.seed(123)
random <- sample(1:nrow(d0))
treat_rows <- random[1:floor(0.5*length(random))]
d0$t <- NA
d0$t[treat_rows] <- 1
d0$t[-treat_rows] <- 0

### Create new var y (realised outcome)
d0$y <- ifelse(d0$t == 1, d0$y1, d0$y0)

# Clean the workspace
rm(random, treat_rows)
```

Before macro

We show the true τ_1

```
mean(d1$y1-d1$y0)
```

```
## [1] 45.62729
```

We can make accurate predictions of τ_1^{PRED}

Using causal match 45.8441494 and using forest 47.1314537.

And here are the important corresponding errors

```
# matching
calc_SE(d1, tauPred_m)
```

```
## [1] 0.04702591
```

```
# forest
calc_SE(d1, tauPredc)
```

```
## [1] 2.262494
```

```
# naive error
calc_NPE(d0, d1)
```

```
## [1] 214.1543
```

Changing the treatment effect function

Now we can change the treatment effect function to accommodate Z

$$\theta_{i,j} = 2 + 10 * x^1 + 1 * Z_j$$

We start by creating a new variable z^1 with values 5 for initial location and 20 for the target location. We do this by replicating the entire data generation process.

```r
# This helper function generates y1
generate_y1 <- function(treatment_function) {
  return(y0 + eval(parse(text  = treatment_function)))
}

treatment_function <- '2 + 10 * x1 + z1'

## Create the synth dataset for D = 0
# Create covariate x1
set.seed(123) # For reproducibility
uniform <- runif(n = 400, min = 0, max = 10)
set.seed(123)
normal <- rnorm(n = 100, mean = 8)
x1 <- c(uniform, normal)
x1 <- x1[x1 <= 10 & x1 >= 0]
# Create y0
y0 <- x1
# Create z1
z1 <- 5
# Define treatment
y1 <- generate_y1(treatment_function)

# Bind together to create simulation.1 d= 0 dataset
d0 <- as.data.frame(cbind(x1, z1, y0, y1))

## Create the synth dataset for D = 1
# Create covariate x1
set.seed(1234) # For reproducibility
uniform <- runif(n = 400, min = 0, max = 10)
set.seed(1234)
normal <- rnorm(n = 100, mean = 2)
x1 <- c(uniform, normal)
x1 <- x1[x1 <= 10 & x1 >= 0]
# Create y0
y0 <- x1
# Create z1
z1 <- 20
# Define treatment
y1 <- generate_y1(treatment_function)

# Bind together to create simulation.1 d= 0 dataset
d1 <- as.data.frame(cbind(x1, z1, y0, y1))

# Clean the workspace
rm(x1, y1, y0, uniform, treatment_function, normal, generate_y1)

### Assign treatment to D = 0

set.seed(123)
random <- sample(1:nrow(d0))
treat_rows <- random[1:floor(0.5*length(random))]
```

```r
d0$t <- NA
d0$t[treat_rows] <- 1
d0$t[-treat_rows] <- 0

### Create new var y (realised outcome)
d0$y <- ifelse(d0$t == 1, d0$y1, d0$y0)

# Clean the workspace
rm(random, treat_rows)
```

Now we can inspect the true ATE first τ_1

```r
mean(d1$y1-d1$y0)
```

```
## [1] 65.62729
```

We also look at NPE which is very low in this case

```r
calc_NPE(d0, d1)
```

```
## [1] 0.1339485
```

Making predictions

```r

tauPred_m_z <- causalMatchFNN(d1, d0, 'x1')

# Using Forest
cf_z <-  causalForest(y ~ x1,
                     data=d0,
                     treatment=d0$t,
                     split.Rule="CT",
                     split.Honest=T,
                     split.Bucket=F,
                     bucketNum = 5,
                     bucketMax = 100,
                     cv.option="CT",
                     cv.Honest=T,
                     minsize = 2L,
                     split.alpha = 0.5,
                     cv.alpha = 0.5,
                     sample.size.total = floor(nrow(d0) / 2),
                     sample.size.train.frac = .5,
                     mtry = ceiling(ncol(d0)/3),
                     nodesize = 3,
                     num.trees= 5,
                     ncolx=length(grep('x', colnames(d0))),
                     ncov_sample= length(grep('x', colnames(d0)))
                     )
```

```
## [1] "Building trees ..."
## [1] "Tree 1"
## [1] 2
## [1] "CT"
## [1] "Tree 2"
## [1] 2
## [1] "CT"
## [1] "Tree 3"
## [1] 2
## [1] "CT"
## [1] "Tree 4"
## [1] 2
## [1] "CT"
## [1] "Tree 5"
## [1] 2
## [1] "CT"
```

```
tauPredc_z <- mean(predict(cf, d1))
```

```
## [1] 498   5
```

SE

```
# Matching
calc_SE(d1, tauPred_m_z)
```

```
## [1] 225.2733
```

```
# Causal Forest
calc_SE(d1, tauPredc_z)
```

```
## [1] 342.0961
```