HA8

Ishwara Hegde

30/11/2020

Introduction

The purpose of this introduction is to give a brief recap of the methods that we will see in the empirical exercise below. Most of the material here is taken from Dmitry's new paper,"Synthetic Difference in Differences". You can check it out (highly recommended) here.

Setup

- We want to estimate the impact of some policy using panel data.
- Policy changes not random – neither across units or time.
- We want to connect connect observed data to unobserved counterfactuals.

Solutions

1. DiD : requires parallel trends and large number of units exposed
2. Synthetic control (SC): small no. of units and no parallel trends.

In the paper linked above, Dmitry et. al combine these two methods and call it Synthetic Difference-in-Differences:

- Like SC, the method re-weights and matches pre-exposure trends to weaken the reliance on parallel trend type assumptions.
- Like DID, it is invariant to additive unit-level shifts.

Mathematical Details

Recall in DiD we obtain the estimates by solving the following TWFE model:

$$\tau^{did}, \hat{\mu}, \hat{\alpha}, \hat{\beta} = \arg\min \sum_{i}^{N} \sum_{t}^{T} (Y_{it} - \mu - \alpha_i - \beta_t - W_{it}\tau)^2$$

In SC we solve the following problem:

$$\tau^{SC}, \hat{\mu}, \hat{\alpha}, \hat{\beta} = \arg\min \{ \sum_{i}^{N} \sum_{t}^{T} (Y_{it} - \mu - \beta_t - W_{it}\tau)^2 \omega_i^{SC} \}$$

Where weights

$$\omega^{SC}$$

align pre-exposure trends in the outcome of unexposed units with those for the exposed units.

Now SDID solves the following:

$$\tau^{SC}, \hat{\mu}, \hat{\alpha}, \hat{\beta} = \arg\min\{\sum_{i}^{N}\sum_{t}^{T}(Y_{it} - \mu - \alpha_i - \beta_t - W_{it}\tau)^2 \omega_i^{SC}\lambda_t^{SDID}\}$$

Notice it introduces two new things:

- With respect to DID, SDID adds the unit and time weights.
- With respect to SC, SDID adds the unit level fixed effects.

  Unit weights are designed so that the average outcome for the treated units are approximately parallel to the averages for control units. Time weights are designed so that, acknowledging that the difference between treated and control averages varies over the pre-treatment period, we adjust for the right pre-treatment difference: the difference during periods that are predictive of what happens after treatment.

Unit fixed effects in applications is found to explain much of the variation therefore its inclusion reduces bias with respect to standard SC.

1. Load and normalize data

```
# Load data
load("cps_data.Rdata")

# Define initial parameters
n <- dim(Y)[1]
T <- dim(Y)[2]
T_0 <- 30

# Normalize Y
Y_norm <- (Y-mean(Y))/sd(Y)
```

2. Construct SVD

```
Y_svd <- svd(Y_norm)

M <- Y_svd$u[,1:4] %*% diag(Y_svd$d[1:4]) %*% t(Y_svd$v[,1:4])
E <- Y_svd$u[,5:T] %*% diag(Y_svd$d[5:T]) %*% t(Y_svd$v[,5:T])

sig2_e <- sum(E^2)/(n*T)

# Confirm M+E equals normalized Y
max(abs(Y_norm-(M+E)))
```

```
## [1] 1.132427e-14
```

3. Decompose M into two components

```
n_ones <- matrix(1, nrow=n, ncol=n)
T_ones <- matrix(1, nrow=T, ncol=T)

F <- (n_ones %*% M/n) + (M %*% T_ones/T)
L <- M-F
```

4. Run a logit of D_i on $u_1 - u_4$ and extract predicted probabilities

```
u <- Y_svd$u[,1:4]
pi <- glm(D~u, family="binomial")$fitted.values
```

5. Define DGP function

```
DGP_1 <- function(n, T, a_0, a_1, F, L, sig2_e, pi) {
  A <- rbinom(n, 1, pi)
  epsilon <- matrix(rnorm(T*n, mean=0, sd=sqrt(sig2_e)), n, T)

  Y <- a_0*F + a_1*L + epsilon
  W <- cbind(matrix(0, nrow=n, ncol=T_0), matrix(1, nrow=n, ncol=T-T_0)) * A

  data_returned <- list("Y" = Y, "W" = W)
  return(data_returned)
}
```

6. Simulate data and compute estimators

```
B <- 400
tau_hat <- matrix(0, nrow = B, ncol = 3)
colnames(tau_hat) <- c("tau_DID", "tau_SC", "tau_SDID")

for (b in 1:B){
  draw <- DGP_1(n, T, 1, 1, F, L, sig2_e, pi)
  rowsort <- order(draw$W[,T])

  data_Y <- draw$Y[rowsort,]
  data_W <- draw$W[rowsort,]

  n_0 <- sum(draw$W[,T]==0)

  tau_hat[b,1] <-      did_estimate(data_Y, N0 = n_0, T0 = T_0)
  tau_hat[b,2] <-       sc_estimate(data_Y, N0 = n_0, T0 = T_0)
  tau_hat[b,3] <- synthdid_estimate(data_Y, N0 = n_0, T0 = T_0)
}
```

7. Plot the distribution of DID and SDID estimators.

```
# Define plotting function
plotting_func <- function(data, x_string){
  p <- ggplot(data=data, mapping=aes(x=x_string)) +
    geom_density(kernel="gaussian", adjust=1.8) +
    geom_vline(xintercept=0, linetype="dotted", color="blue") +
    theme_minimal()
  return(p)
}

# Create the plots for DGP 1
tau_hat <- data.table(tau_hat)
p1 <- plotting_func(data = tau_hat, x_string = tau_hat$tau_DID) +
        xlab("Diff-in-Diff") + xlim(-.15,.15)
```

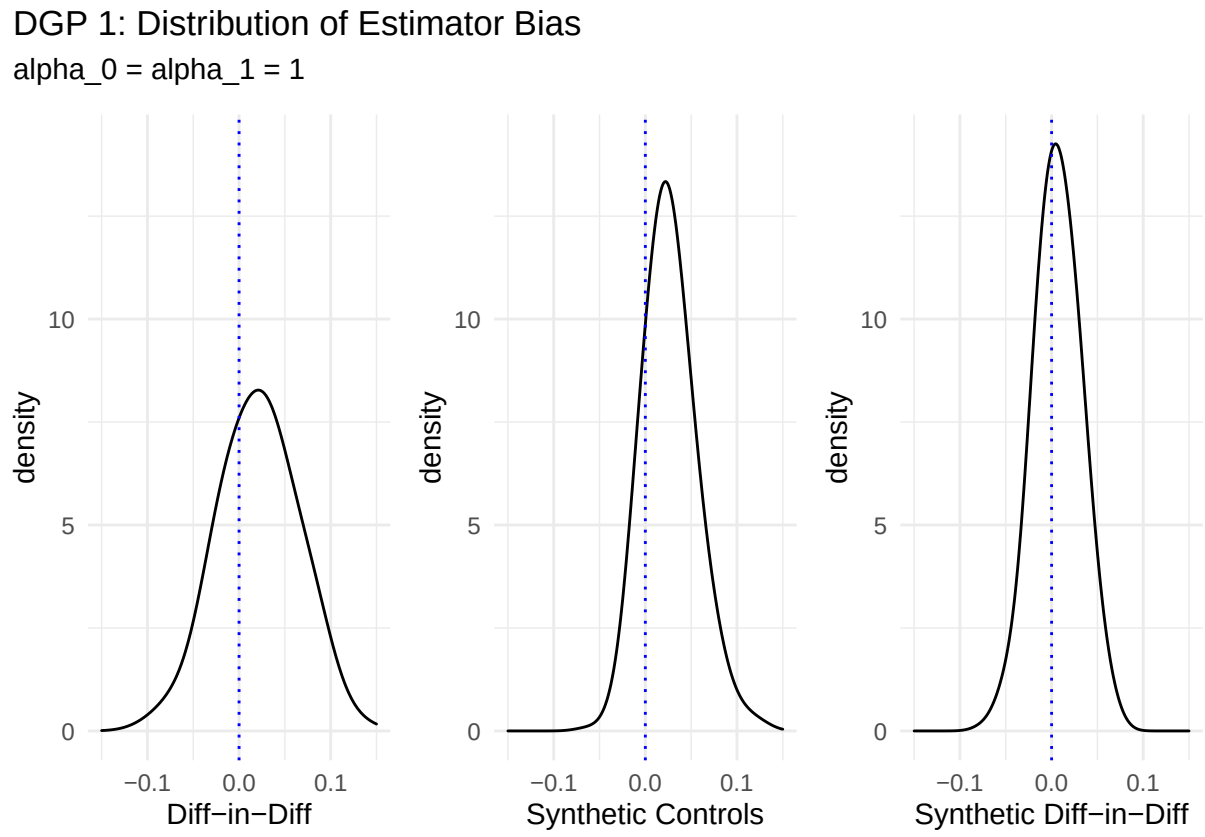
```
p2 <- plotting_func(data = tau_hat, x_string = tau_hat$tau_SC) +
        xlab("Synthetic Controls") + xlim(-.15,.15)
p3 <- plotting_func(data = tau_hat, x_string = tau_hat$tau_SDID) +
        xlab("Synthetic Diff-in-Diff") + xlim(-.15,.15)

p_dgp1 <- p1 + p2 + p3 + plot_annotation(
  title = "DGP 1: Distribution of Estimator Bias",
  subtitle = "alpha_0 = alpha_1 = 1"
)

p_ranges_x <- c(ggplot_build(p_dgp1[[1]])$layout$panel_scales_x[[1]]$range$range,
  ggplot_build(p_dgp1[[2]])$layout$panel_scales_x[[1]]$range$range,
  ggplot_build(p_dgp1[[3]])$layout$panel_scales_x[[1]]$range$range)

p_ranges_y <-c(ggplot_build(p_dgp1[[1]])$layout$panel_scales_y[[1]]$range$range,
        ggplot_build(p_dgp1[[2]])$layout$panel_scales_y[[1]]$range$range,
        ggplot_build(p_dgp1[[3]])$layout$panel_scales_y[[1]]$range$range)


p_dgp1 &
  xlim(min(p_ranges_x), max(p_ranges_x)) &
  ylim(min(p_ranges_y), max(p_ranges_y))
```



It is clear that the bias of the diff-in-diff estimator is larger both in terms of its mean and its variance. Allowing the weights assigned to each data point to vary in both dimensions helps to correct for the fact that our treated subsample does not follow perfect parallel trends with the untreated subsample. It also helps to

correct for imbalances in the propensity for treatment across the treated and untreated subsamples.

8. Repeat DGP and estimation for alternative alpha parametrizations.

```
tau_hat2 <- matrix(0, nrow = B, ncol = 3)
tau_hat3 <- matrix(0, nrow = B, ncol = 3)
colnames(tau_hat2) <- c("tau_DID", "tau_SC", "tau_SDID")
colnames(tau_hat3) <- c("tau_DID", "tau_SC", "tau_SDID")

for (b in 1:B){
  # Alternative Draw #1 - alpha_0 = 1, alpha_1 = 0
  draw <- DGP_1(n, T, 1, 0, F, L, sig2_e, pi)

  rowsort <- order(draw$W[,T])
  data_Y <- draw$Y[rowsort,]
  data_W <- draw$W[rowsort,]

  n_0 <- sum(draw$W[,T]==0)

  tau_hat2[b,1] <-      did_estimate(data_Y, N0 = n_0, T0 = T_0)
  tau_hat2[b,2] <-       sc_estimate(data_Y, N0 = n_0, T0 = T_0)
  tau_hat2[b,3] <- synthdid_estimate(data_Y, N0 = n_0, T0 = T_0)

  # Alternative Draw #2 - alpha_0 = 0, alpha_1 = 1
  draw <- DGP_1(n, T, 0, 1, F, L, sig2_e, pi)

  rowsort <- order(draw$W[,T])
  data_Y <- draw$Y[rowsort,]
  data_W <- draw$W[rowsort,]

  n_0 <- sum(draw$W[,T]==0)

  tau_hat3[b,1] <-      did_estimate(data_Y, N0 = n_0, T0 = T_0)
  tau_hat3[b,2] <-       sc_estimate(data_Y, N0 = n_0, T0 = T_0)
  tau_hat3[b,3] <- synthdid_estimate(data_Y, N0 = n_0, T0 = T_0)
}

summary(tau_hat)
```

```
##     tau_DID              tau_SC             tau_SDID
##  Min.   :-0.107394   Min.   :-0.062916   Min.   :-0.075108
##  1st Qu.:-0.008465   1st Qu.: 0.007496   1st Qu.:-0.010724
##  Median : 0.021430   Median : 0.024586   Median : 0.005846
##  Mean   : 0.020822   Mean   : 0.026584   Mean   : 0.006308
##  3rd Qu.: 0.048370   3rd Qu.: 0.044269   3rd Qu.: 0.024027
##  Max.   : 0.143706   Max.   : 0.130365   Max.   : 0.069954
```

```
summary(tau_hat2)
```

```
##     tau_DID              tau_SC             tau_SDID
##  Min.   :-0.0295256   Min.   :-0.026754   Min.   :-0.0333660
##  1st Qu.:-0.0074298   1st Qu.:-0.003324   1st Qu.:-0.0095168
##  Median :-0.0001467   Median : 0.005173   Median :-0.0007284
##  Mean   : 0.0004087   Mean   : 0.006339   Mean   : 0.0004167
##  3rd Qu.: 0.0079169   3rd Qu.: 0.013702   3rd Qu.: 0.0091805
```

```
##  Max.   : 0.0430936   Max.   : 0.091893   Max.   : 0.0541877
```

```
summary(tau_hat3)
```

```
##     tau_DID             tau_SC             tau_SDID
##  Min.   :-0.11788   Min.   :-0.050878   Min.   :-0.061881
##  1st Qu.:-0.00570   1st Qu.:-0.008903   1st Qu.:-0.010099
##  Median : 0.02491   Median : 0.003801   Median : 0.006059
##  Mean   : 0.01995   Mean   : 0.004196   Mean   : 0.006025
##  3rd Qu.: 0.04749   3rd Qu.: 0.017212   3rd Qu.: 0.023477
##  Max.   : 0.12825   Max.   : 0.062167   Max.   : 0.101490
```

Each parametrization helps to reveal a bit more about the bias issues in each estimation strategy. First, we should understand the interpretation of α_0 and α_1. α_0 determines whether or not the data draws include $\mathbf{F}$, which in our simulated data is like a two-way fixed effect component (summing the time-mean and unit-mean of $\mathbf{M}$). α_1 does the same for $\mathbf{L} = \mathbf{M} - \mathbf{F}$, which contains the remaining unit-time interaction component. Note that $\mathbf{M}$ contains the four principal components of the SVD of $\tilde{\mathbf{Y}}$, which we can think of as the systematic component of the data (with $\mathbf{E}$ being the noise).

When $\alpha_0 = \alpha_1 = 1$, the synthetic diff-in-diff estimator is the least biased and most efficient.

When $\alpha_0 = 1$ and $\alpha_1 = 0$, the regulator diff-in-diff estimator is the best estimator but is only marginally better than synthetic diff-in-diff. This is because only the fixed effect component is included in the data generation, so the parallel trends assumption holds by construction.

When $\alpha_0 = 0$ and $\alpha_1 = 1$, the synthetic controls estimator is the best but again is only marginally better than synthetic diff-in-diff.

9. Repeat parts 5-7 but with an alternative DGP

```
# Define new DGP function (using a constant average probability instead of
#individual predictions)
DGP_2 <- function(n, T, a_0, a_1, F, L, sig2_e, pi) {

  pibar <- mean(pi)

  A <- rbinom(n, 1, pibar)
  epsilon <- matrix(rnorm(T*n, mean=0, sd = sqrt(sig2_e)), n, T)

  Y <- a_0*F + a_1*L + epsilon
  W <- cbind(matrix(0, nrow=n, ncol=T_0), matrix(1, nrow=n, ncol=T-T_0)) * A

  data_returned <- list("Y" = Y, "W" = W)
  return(data_returned)
}
```

```
# Run the DGP simulation and estimation process again
tau_hat4 <- matrix(0, nrow = B, ncol = 3)
colnames(tau_hat4) <- c("tau_DID", "tau_SC", "tau_SDID")

for (b in 1:B){
  draw <- DGP_2(n, T, 1, 1, F, L, sig2_e, pi)
  rowsort <- order(draw$W[,T])

  data_Y <- draw$Y[rowsort,]
  data_W <- draw$W[rowsort,]
```

```
  n_0 <- sum(draw$W[,T]==0)

  tau_hat4[b,1] <-      did_estimate(data_Y, N0 = n_0, T0 = T_0)
  tau_hat4[b,2] <-       sc_estimate(data_Y, N0 = n_0, T0 = T_0)
  tau_hat4[b,3] <- synthdid_estimate(data_Y, N0 = n_0, T0 = T_0)
}
```

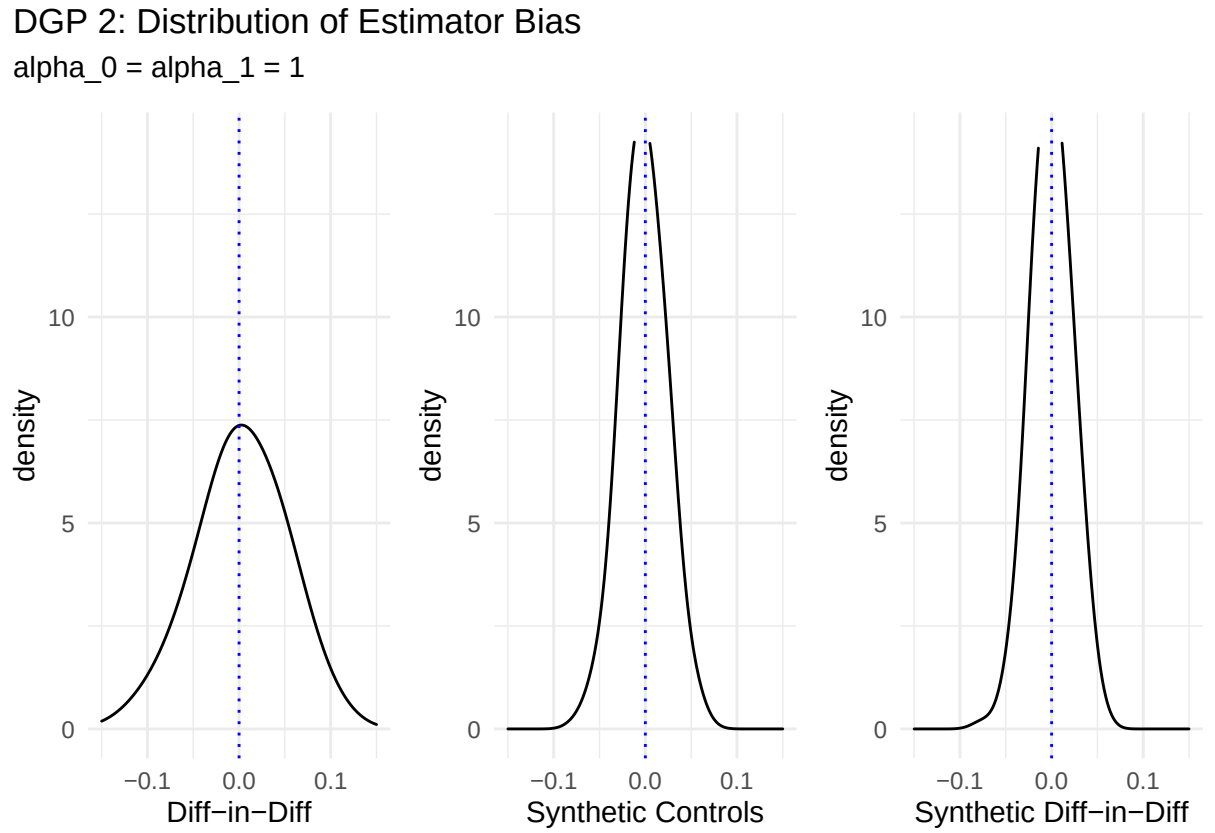
```
# Create the plots for DGP 2
tau_hat4 <- data.table(tau_hat4)
p4 <- plotting_func(data=tau_hat4,x_string = tau_hat4$tau_DID) +
        xlab("Diff-in-Diff") + xlim(-.15,.15)
p5 <- plotting_func(data=tau_hat4,x_string = tau_hat4$tau_SC) +
        xlab("Synthetic Controls") + xlim(-.15,.15)
p6 <- plotting_func(data=tau_hat4,x_string = tau_hat4$tau_SDID) +
        xlab("Synthetic Diff-in-Diff") + xlim(-.15,.15)

p_dgp2 <- p4 + p5 + p6 + plot_annotation(
  title = "DGP 2: Distribution of Estimator Bias",
  subtitle= "alpha_0 = alpha_1 = 1"
)

p_ranges_x_2<- c(ggplot_build(p_dgp2[[1]])$layout$panel_scales_x[[1]]$range$range,
  ggplot_build(p_dgp1[[2]])$layout$panel_scales_x[[1]]$range$range,
  ggplot_build(p_dgp1[[3]])$layout$panel_scales_x[[1]]$range$range)

p_ranges_y_2<-c(ggplot_build(p_dgp2[[1]])$layout$panel_scales_y[[1]]$range$range,
        ggplot_build(p_dgp1[[2]])$layout$panel_scales_y[[1]]$range$range,
        ggplot_build(p_dgp1[[3]])$layout$panel_scales_y[[1]]$range$range)


p_dgp2 &
  xlim(min(p_ranges_x_2), max(p_ranges_x_2)) &
  ylim(min(p_ranges_y_2), max(p_ranges_y_2))
```



Under the new DGP, the propensity for treatment is identical across all observations. That is, assignment to treatment is actually random. Here, the bias issue for the diff-in-diff estimator largely vanishes. However, the synthetic diff-in-diff estimator remains more efficient, just as it was for the original DGP with heterogeneity in treatment propensity.