Simulation results

Xiaodan Lyu, Emily Berg, Heike Hofmann

4/6/2020

```
rm(list = ls(all = T))
```

Table 1

```
load("intermediate_results/Cor0.9_D60_nsim1000_seed2020_Result.RData")
mcmse.eb <- tapply(colMeans((pred.eb.store-YbarNis.store)^2), nis, mean)
## MC MSE of the proposed EB predictor
sprintf("%.2f", mcmse.eb*10^5)
```

```
## [1] "24.09" "15.92" "9.27"
```

```
## MSE differences between the alternative predictors and the EB predictor
dmse.alt <- function(pred.store){
  mcmse <- apply((pred.store-YbarNis.store)^2, 1, tapply, nis, mean)
  dmse <- mcmse-as.vector(mcmse.eb)
  diff <- rowMeans(dmse)*1e+5
  err <- 1.96*sqrt(apply(dmse, 1, var)/1000)*1e+5
  paste0(sprintf("%.2f", diff), " (", sprintf("%.2f", err), ")")
}
dmse.result <- do.call(
  "cbind",
  lapply(list(eb0 = pred.eb0.store,
              pi = pred.pi.store,
              zi = pred.zi.store,
              si = pred.si.store),
         dmse.alt))
dmse.result
```

```
##      eb0           pi            zi            si
## [1,] "1.17 (0.99)" "1.20 (0.99)" "4.28 (0.96)" "71.58 (7.32)"
## [2,] "0.89 (0.72)" "0.90 (0.72)" "3.57 (0.70)" "78.04 (8.17)"
## [3,] "0.64 (0.35)" "0.63 (0.35)" "2.76 (0.36)" "71.29 (6.54)"
```

Table 2

```r
## MSE differences between the EB(0) predictor and the EB predictor
dmse.store <- c()
rhos <- c(-0.9, -0.6, -0.3, 0, 0.3, 0.6, 0.9)
for (rho in rhos){
  load(sprintf("intermediate_results/Cor%s_D60_nsim1000_seed2020_Result.RData", rho))
  mcmse.eb <- apply((pred.eb.store-YbarNis.store)^2, 1, tapply, nis, mean)
  mcmse.eb0 <- apply((pred.eb0.store-YbarNis.store)^2, 1, tapply, nis, mean)
  dmse <- mcmse.eb0-mcmse.eb
  diff <- rowMeans(dmse)*1e+5
  err <- 1.96*sqrt(apply(dmse, 1, var)/1000)*1e+5
  dmse.store <- cbind(dmse.store, paste0(sprintf("%.2f", diff), " (", sprintf("%.2f", err), ")"))
}
colnames(dmse.store) <- rhos
dmse.store
```

```
##      -0.9          -0.6          -0.3          0             0.3
## [1,] "2.43 (0.25)" "0.97 (0.18)" "0.16 (0.12)" "-0.14 (0.08)" "-0.05 (0.09)"
## [2,] "2.49 (0.23)" "0.79 (0.16)" "0.01 (0.13)" "-0.14 (0.08)" "-0.03 (0.07)"
## [3,] "1.54 (0.13)" "0.39 (0.10)" "-0.02 (0.06)" "-0.08 (0.04)" "0.02 (0.04)"
##      0.6           0.9
## [1,] "0.45 (0.13)" "1.17 (0.19)"
## [2,] "0.27 (0.12)" "0.89 (0.16)"
## [3,] "0.23 (0.06)" "0.64 (0.09)"
```

Table 3

```r
## assemble simulation results
seeds <- 2015:2019
m12boot <- m1biasboot <- m2boot <- mseboot <- c()
ybar <- predeb <- onestep <- c()
for (seed in seeds){
  load(sprintf("intermediate_results/bootstrap/Cor0.9_D60_nsim200_seed%.0f_Result.RData", seed))
  m12boot <- rbind(m12boot, M12boot.store)
  m1biasboot <- rbind(m1biasboot, M1biasboot.store)
  m2boot <- rbind(m2boot, M2boot.store)
  mseboot <- rbind(mseboot, MSEboot.store)
  ybar <- rbind(ybar, YbarNis.store)
  predeb <- rbind(predeb, pred.eb.store)
  onestep <- rbind(onestep, mse.eb.store)
}
## bias and coverage for the MSE estimators
mcmse <- colMeans((predeb - ybar)^2)
evalmse <- function(estmse){
  bmse <- apply(estmse, 1, function(x) tapply(x-mcmse, nis, mean))
  cp <- apply(abs(predeb - ybar) <= 1.96*sqrt(estmse), 1, tapply, nis, mean)
  bmse.val <- rowMeans(bmse)*1e+5
  bmse.err <- 1.96*sqrt(apply(bmse, 1, var)/1000)*1e+5
  cp.val <- rowMeans(cp)*100
  cp.err <- 1.96*sqrt(apply(cp, 1, var)/1000)*100
  data.frame(
    bmse = paste0(sprintf("%.2f", bmse.val), " (", sprintf("%.2f", bmse.err), ")"),
```

```
    cp = paste0(sprintf("%.2f", cp.val), " (", sprintf("%.2f", cp.err), ")"))
}
output <- do.call("cbind", lapply(list(
  onestep = onestep, boot = mseboot, semiboot = onestep + m2boot),
  evalmse))
output
```

```
##   onestep.bmse   onestep.cp    boot.bmse      boot.cp semiboot.bmse
## 1 -0.66 (0.51) 94.94 (0.33)  0.48 (0.48) 94.54 (0.33)   0.46 (0.53)
## 2 -0.96 (0.32) 94.88 (0.32) -0.00 (0.30) 94.08 (0.33)  -0.14 (0.33)
## 3 -0.68 (0.19) 94.47 (0.35) -0.16 (0.16) 93.78 (0.32)  -0.12 (0.19)
##    semiboot.cp
## 1 95.36 (0.31)
## 2 95.48 (0.30)
## 3 95.37 (0.31)
```

Table S2

```
## composition of the three MSE components
bootres <- do.call("cbind", lapply(
  list(m2boot/mseboot, m1biasboot/mseboot, m12boot/mseboot), colMeans))
bootres <- apply(bootres, 2, tapply, nis, mean)
round(bootres*100, 2)
```

```
##    [,1]  [,2]  [,3]
## 5  4.93 -2.14 -0.14
## 10 5.65 -2.98 -0.21
## 20 6.38 -3.06 -0.26
```