Power Analysis on Shuttle dataset

Significance level 0.2

2023-05-16

```r
library(doSNOW)
library(foreach)
library(nout)
library(tictoc)
library(isotree)
library(readr)
library(R.matlab)


compact_results = function(res){
  resT=as.data.frame(t(res))

  discoveries = as.data.frame(cbind("d_BH"=unlist(resT$d_BH),
                                   "d_StoBH"=unlist(resT$d_StoBH),
                                   "d_Sim"=unlist(resT$d_Sim),
                                   "d_StoSimes"=unlist(resT$d_StoSimes),
                                   "d_WMW"=unlist(resT$d_WMW)))
  mean.discoveries = apply(discoveries, MARGIN = 2, FUN = mean)


  power.GlobalNull = as.data.frame(discoveries>0)
  mean.powerGlobalNull = apply(power.GlobalNull, MARGIN = 2, FUN = mean)

  return(list("discoveries" = discoveries,
            "mean.discoveries" = mean.discoveries,
            "power.GlobalNull" = power.GlobalNull,
            "mean.powerGlobalNull" = mean.powerGlobalNull,
            "pi.not" = unlist(resT$pi.not),
            "uniques"=unlist(resT$uniques),
            "n1"=unlist(resT$n1),
            "alpha"=unlist(resT$alpha)))
}
```

The aim is to compare on Shuttle datasets the performance of three closed testing procedures, which respectively use Simes local test with and without Storey estimator for the proportion of true null hypotheses and Wilcoxon-Mann-Whitney local test.

We fix the train set on which we train the isolation forest algorithm and we generate $B = 10^4$ calibration and test sets. For each $b = 1, \dots, B$ we compute the number of discoveries obtained by Benjamini-Hochberg procedure with and without Storey's estimator for the proportion of true null hypotheses, by closed testing using Simes local test with and without Storey's estimator and by closed testing using Wilcoxon-Mann-Whitney local test.

Shuttle (Statlog) dataset

Shuttle dataset (available at http://odds.cs.stonybrook.edu/shuttle-dataset) consists of 49097 observations, among which $n_{inliers} = 45586$ items are inliers and the remaining $n_{outliers} = 3511$ are outliers. We will denote by l, m, n respectively the train set, the calibration set and the test set size. And reproducing the same setting as in [1], we have that $m + l = n_{inliers}/2$, $m = min\{2000, l/2\}$ and $n = min\{2000, l/3\}$. Moreover, in order to have exact control of type I errors at the significance level $\alpha = 0.2$. we require $\alpha = n/(m + 1)$. In the case of Shuttle dataset we obtain $l = 12794, \quad m = 9999, \quad n = 2000$.

Load the data and set the parameters as described above.

```
data = readMat("~/nout/trials/RealData/Datasets/Dataset shuttle/shuttle.mat")
dataset = cbind(data$X, data$y); colnames(dataset)[ncol(dataset)] = "y"
out_ind = which(dataset[,ncol(dataset)]==1)
in_ind = which(dataset[,ncol(dataset)]==0)

# Initializing parameters
set.seed(321)

B=10^4

l = 12794
m = 9999
n = 2000
myalpha = n/(m+1)

tr_ind = sample(in_ind, size = l)
in_ind2 = setdiff(in_ind, tr_ind)
tr = dataset[tr_ind,]
n_cpus = parallel::detectCores()
iso.fo = isotree::isolation.forest(tr, ndim = ncol(dataset), ntrees = 150, sample_size = 256,
                                   nthreads = n_cpus, scoring_metric = "depth",
                                   output_score = TRUE)
isofo.model = iso.fo$model
mycrit = nout::critWMW(m = n, n = m, alpha = myalpha)
```

All inliers

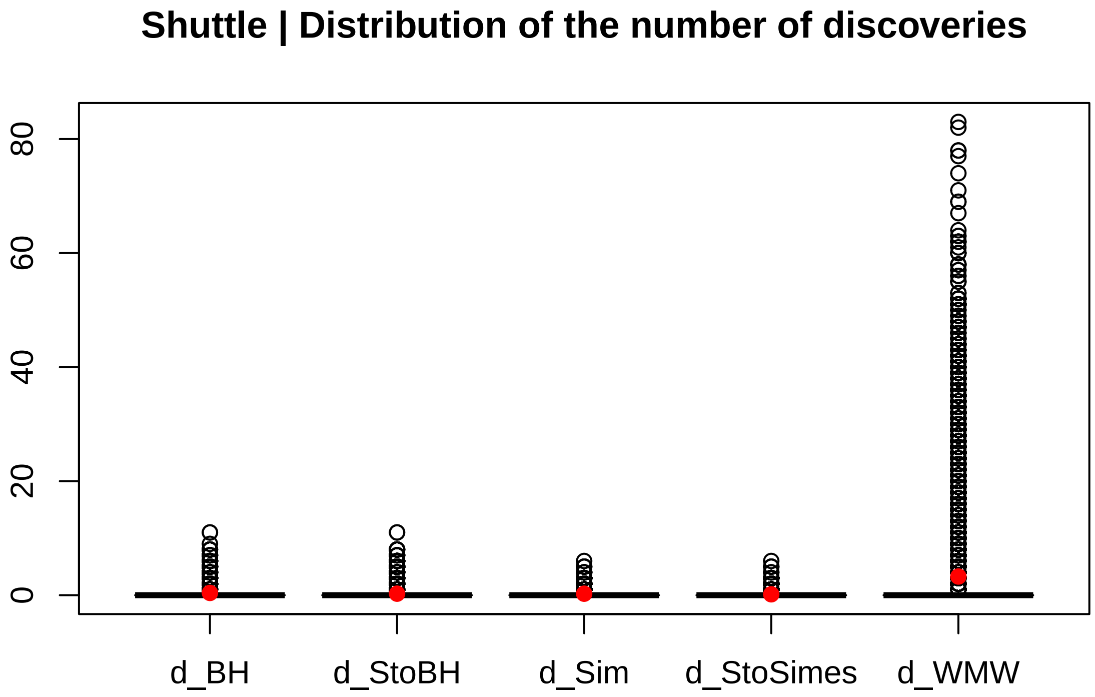
We now set the proportion of inliers equal to 1, so that the number of outliers $n_1 = 0$.

```
n1=0

cl <- makeCluster(parallel::detectCores())
clusterEvalQ(cl, {library(isotree)})
```

```
## [[1]]
## [1] "isotree"   "snow"      "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "methods"   "base"
##
## [[2]]
## [1] "isotree"   "snow"      "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "methods"   "base"
##
## [[3]]
## [1] "isotree"   "snow"      "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "methods"   "base"
```

```
##
## [[4]]
## [1] "isotree"   "snow"      "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "methods"   "base"
```

```
registerDoSNOW(cl)


res = foreach(b = 1:B, .combine=cbind) %dopar% {
  n0 = n - n1
  N = n0 + m
  in_index3 = sample(in_ind, size = N)
  cal_ind = in_index3[1:m]
  te_ind = in_index3[(m + 1):N]
  cal = dataset[cal_ind,]
  te = dataset[te_ind,]
  S_cal = isotree::predict.isolation_forest(isofo.model, cal, type = "score")
  S_te = isotree::predict.isolation_forest(isofo.model, te, type = "score")
  d_WMW = nout::d_mannwhitney(S_Y = S_te, S_X = S_cal, crit = mycrit)
  d_Sim = nout::d_Simes(S_X = S_cal, S_Y = S_te, alpha = myalpha)
  StoSimes = nout::d_StoreySimes(S_X = S_cal, S_Y = S_te, alpha = myalpha)
  d_StoSimes = StoSimes$d
  pi.not = StoSimes$pi.not
  d_BH = nout::d_benjhoch(S_X = S_cal, S_Y = S_te, alpha = myalpha)
  d_StoBH = nout::d_StoreyBH(S_X = S_cal, S_Y = S_te, alpha = myalpha)
  uniques = length(unique(c(S_cal, S_te)))
  return(list("d_BH" = d_BH,
              "d_StoBH" = d_StoBH,
              "d_Sim" = d_Sim,
              "d_StoSimes" = d_StoSimes,
              "d_WMW" = d_WMW,
              "uniques" = uniques,
              "n1" = n1,
              "pi.not" = pi.not,
              "alpha" = myalpha))
}

stopCluster(cl)


results = compact_results(res)

boxplot(results$discoveries, main="Shuttle | Distribution of the number of discoveries")
points(x=1:5, y=results$mean.discoveries, pch=19, col="red")
```



```
results$mean.discoveries
```

```
##       d_BH    d_StoBH      d_Sim d_StoSimes      d_WMW
##     0.3898     0.2596     0.2553     0.1529     3.2692
```

```
results$mean.powerGlobalNull
```

```
##       d_BH    d_StoBH      d_Sim d_StoSimes      d_WMW
##     0.2034     0.1211     0.2034     0.1211     0.2002
```

```
resShuttle0 = results
save(resShuttle0,
     file="~/nout/trials/RealData/PowerStudy/New!/alpha0.2/ShuttleOnly0.2/resShuttle0")
```

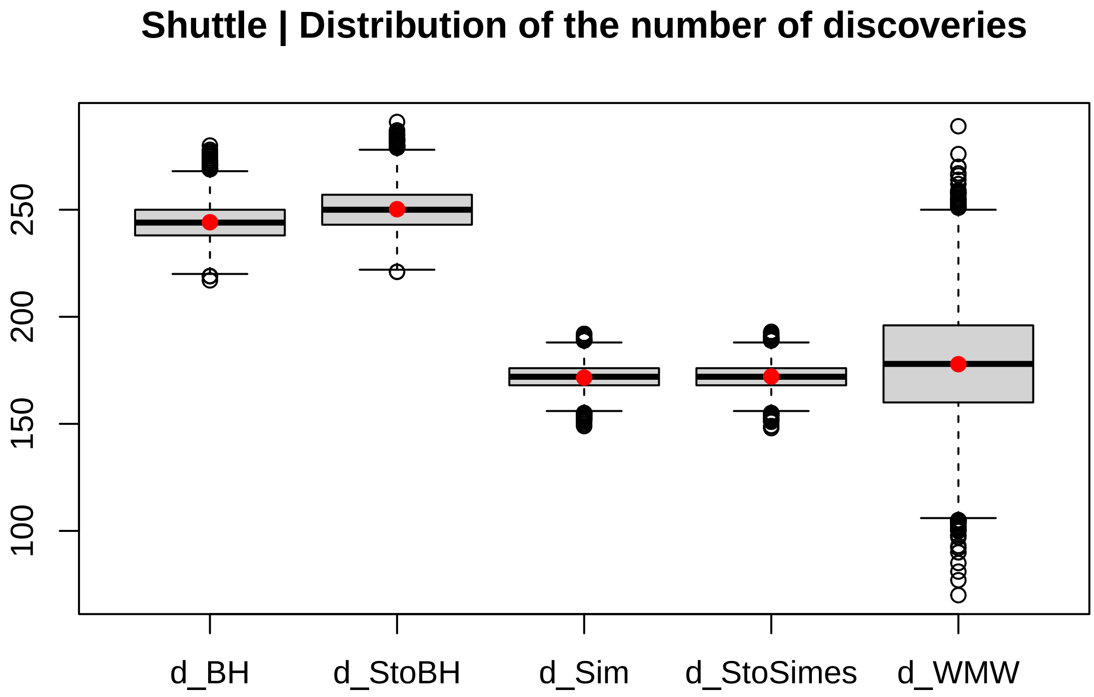
10% outliers

We now set the proportion of inliers equal to 0.9. Referring to Digits dataset we have that the number of inliers is $n_0 = 1800$ and the number of outliers is $n_1 = 200$.

```
n1=round(0.1*n)

cl <- makeCluster(parallel::detectCores())
clusterEvalQ(cl, {library(isotree)})
```

```
## [[1]]
## [1] "isotree"   "snow"      "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "methods"   "base"
##
## [[2]]
## [1] "isotree"   "snow"      "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "methods"   "base"
##
## [[3]]
## [1] "isotree"   "snow"      "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "methods"   "base"
##
## [[4]]
## [1] "isotree"   "snow"      "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "methods"   "base"
```

```
registerDoSNOW(cl)

res = foreach(b = 1:B, .combine=cbind) %dopar% {
  n0 = n - n1
  N = n0 + m
  in_index3 = sample(in_ind, size = N)
  cal_ind = in_index3[1:m]
  tein_ind = in_index3[(m + 1):N]
  teout_ind = sample(out_ind, size = n1)
  cal = dataset[cal_ind,]
  te = dataset[c(tein_ind, teout_ind),]
  S_cal = predict.isolation_forest(isofo.model, cal, type = "score")
  S_te = predict.isolation_forest(isofo.model, te, type = "score")
  d_WMW = nout::d_mannwhitney(S_Y = S_te, S_X = S_cal, crit = mycrit)
  d_Sim = nout::d_Simes(S_X = S_cal, S_Y = S_te, alpha = myalpha)
  StoSimes = nout::d_StoreySimes(S_X = S_cal, S_Y = S_te, alpha = myalpha)
  d_StoSimes = StoSimes$d
  pi.not = StoSimes$pi.not
  d_BH = nout::d_benjhoch(S_X = S_cal, S_Y = S_te, alpha = myalpha)
  d_StoBH = nout::d_StoreyBH(S_X = S_cal, S_Y = S_te, alpha = myalpha)
  uniques = length(unique(c(S_cal, S_te)))
  return(list("d_BH" = d_BH,
              "d_StoBH" = d_StoBH,
              "d_Sim" = d_Sim,
              "d_StoSimes" = d_StoSimes,
              "d_WMW" = d_WMW,
              "uniques" = uniques,
              "n1" = n1,
              "pi.not" = pi.not,
              "alpha" = myalpha))

}

stopCluster(cl)

results = compact_results(res)

boxplot(results$discoveries, main="Shuttle | Distribution of the number of discoveries")
points(x=1:5, y=results$mean.discoveries, pch=19, col="red")
```



```
results$mean.discoveries
```

```
##       d_BH    d_StoBH      d_Sim d_StoSimes      d_WMW
##   244.1546   250.2530   171.5925   172.0958   177.8617
```

```
results$mean.powerGlobalNull
```

```
##       d_BH    d_StoBH      d_Sim d_StoSimes      d_WMW
##          1          1          1          1          1
```

```
resShuttle10 = results
save(resShuttle10,
     file="~/nout/trials/RealData/PowerStudy/New!/alpha0.2/ShuttleOnly0.2/resShuttle10")
```