Real Data Power Analysis

Significance level 0.2

2023-05-14

```r
library(nout)
library(R.matlab)
library(isotree)
library(tictoc)
```

The aim is to compare on Digits, Credit Card and 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.

Digits dataset

Digits dataset (available at http://odds.cs.stonybrook.edu/pendigits-dataset) consists of 6870 observations, among which $n_{inliers} = 6714$ items are inliers and the remaining $n_{outliers} = 156$ are outliers. We will denote by n, l, m respectively the train set, the calibration set and the test set size. And reproducing the same setting as in [1], we have that $m + n = n_{inliers}/2$, $m = min\{2000, l/2\}$ and $n = min\{2000, l/3\}$. In the case of Digits dataset we obtain

$$m + n = 6714/2, \quad m = l/3, \quad n = l/3$$

from which $n = 2517.75, \quad l = 839.25, \quad m = 839.25$. Arbitrarily, we choose to set

$$n = 1258, \quad l = 2099, \quad m = 420$$

in order to have exact control of type I errors at the significance level $\alpha = 0.2$.

Load the data and set the parameters as described above.

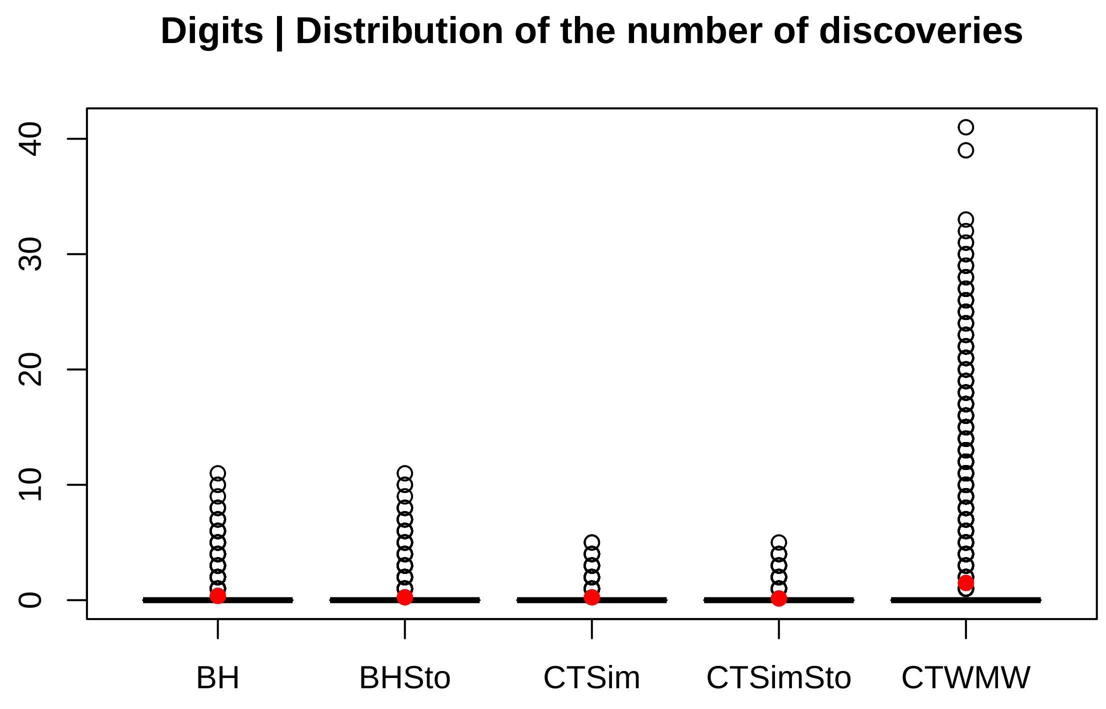
```r
set.seed(321)

# Initializing parameters
l = 2518
m = 2099
n = 420
myalpha = n/(m+1)

data = readMat("G:\\Il mio Drive\\PHD\\Progetto di ricerca\\Conformal Inference Project\\Simulazioni\\7
dataset = cbind(data$X, data$y); colnames(dataset)[ncol(dataset)] = "y"
in_ind = which(dataset[,ncol(dataset)]==0)
out_ind = which(dataset[,ncol(dataset)]==1)
```

All inliers

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

```r
B=10^4

n1=0
n0=n-n1
N=m+n
tr_ind = sample(in_ind, size = l)
tr = dataset[tr_ind,]
iso.fo = isolation.forest(tr, ndim=ncol(dataset), ntrees=100, nthreads=parallel::detectCores(),
                          scoring_metric = "depth", output_score = TRUE)
in_index2 = setdiff(in_ind, tr_ind)
mycrit = nout::critWMW(m=n,n=m,alpha=myalpha)

d_WMW = rep(0,B)
d_Simes = rep(0,B)
d_StoSimes = rep(0,B)
d_BH = rep(0,B)
d_StoBH = rep(0,B)
uniques = rep(0,B)

for (b in 1:B){
  cal_ind = sample(in_index2, size = m)
  in_index3 = setdiff(in_index2, cal_ind)
  te_ind = sample(in_index3, size = n)
  cal = dataset[cal_ind,]
  te = dataset[te_ind,]

  S_cal = predict.isolation_forest(iso.fo$model, cal, type = "score")
  S_te = predict.isolation_forest(iso.fo$model, te, type = "score")
  #U_i = sapply(1:n, function(i) sum(S_te[i]>S_cal))
  #U = sum(U_i)

  uniques[b] = length(unique(c(S_cal, S_te)))
  d_WMW[b] = nout::d_mannwhitney(S_Y=S_te, S_X=S_cal, crit = mycrit)
  #resU[b] = U >=mycrit$crit.vals[1]
  d_Simes[b] = nout::d_Simes(S_X=S_cal, S_Y=S_te, alpha=myalpha)
  d_StoSimes[b] = nout::d_StoreySimes(S_X=S_cal, S_Y=S_te, alpha=myalpha)$d
  d_BH[b] = nout::d_benjhoch(S_X=S_cal, S_Y=S_te, alpha=myalpha)
  d_StoBH[b] = nout::d_StoreyBH(S_X=S_cal, S_Y=S_te, alpha=myalpha)
}


discoveries = as.data.frame(cbind("d_BH"=d_BH, "d_StoBH"=d_StoBH, "d_Simes"=d_Simes,
                                  "d_StoSimes"=d_StoSimes, "d_WMW"=d_WMW))
colnames(discoveries) = c("BH", "BHSto", "CTSim", "CTSimSto", "CTWMW")
mean.discoveries = apply(discoveries, MARGIN = 2, FUN = mean)

powerGlobalNull = as.data.frame(cbind("d_BH"=d_BH>0, "d_StoBH"=d_StoBH>0, "d_Simes"=d_Simes>0,
                                      "d_StoSimes"=d_StoSimes>0, "d_WMW"=d_WMW>0))
colnames(powerGlobalNull) = c("BH", "BHSto", "CTSim", "CTSimSto", "CTWMW")
mean.powerGlobalNull = apply(powerGlobalNull, MARGIN = 2, FUN = mean)
```

```
prova.d_WMW = mean(d_WMW>0)
#prova.resU = mean(resU)


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



```
mean.discoveries
```

```
##       BH    BHSto    CTSim CTSimSto    CTWMW
##   0.3740   0.2505   0.2511   0.1522   1.4941
```

```
mean.powerGlobalNull
```

```
##       BH    BHSto    CTSim CTSimSto    CTWMW
##   0.2025   0.1216   0.2025   0.1216   0.1893
```

```
prova.d_WMW
```

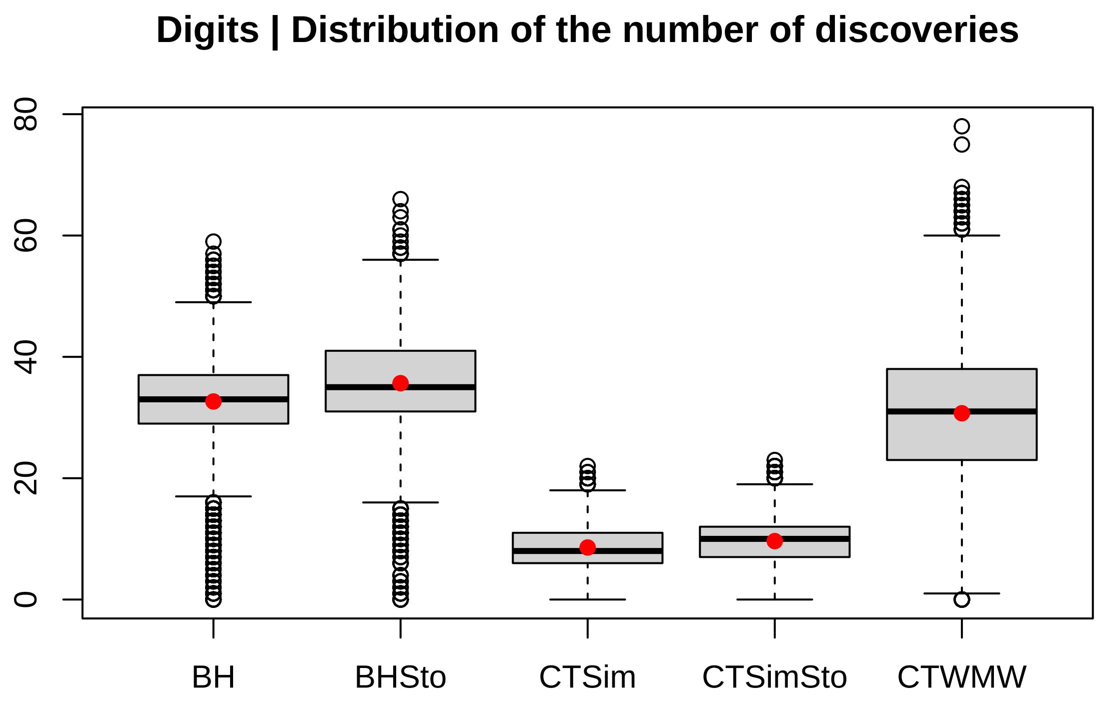
```
## [1] 0.1893
```

```
#prova.resU

res=list("prova.d_WMW"=prova.d_WMW,
         #"prova.resU"=prova.resU,
         #"resU" = resU,
         "uniques"=uniques,
         "discoveries"=discoveries,
         "mean.discoveries" = mean.discoveries,
         "powerGlobalNull"=powerGlobalNull,
         "mean.powerGlobalNull"=mean.powerGlobalNull,
         "n1"=n1,
         "alpha"=myalpha)

resDigits0 = res
save(resDigits0,
     file="C:/Users/c.magnani9/Documents/nout/trials/RealData/PowerStudy/New&TidyNoSimuFunction/alpha0.
```

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 $n_0 = 378$ and the number of outliers is $n_1 = 42$.

```
B=10^4

n1=round(0.1*n)
n0=n-n1
N=m+n
tr_ind = sample(in_ind, size = l)
tr = dataset[tr_ind,]
iso.fo = isolation.forest(tr, ndim=ncol(dataset), ntrees=100, nthreads=parallel::detectCores()-1L,
                          scoring_metric = "depth", output_score = TRUE)
in_index2 = setdiff(in_ind, tr_ind)
mycrit = nout::critWMW(m=n,n=m,alpha=myalpha)

d_WMW = rep(0,B)
d_Simes = rep(0,B)
d_StoSimes = rep(0,B)
d_BH = rep(0,B)
d_StoBH = rep(0,B)
uniques = rep(0,B)

for (b in 1:B){
  in_index3 = sample(in_index2, size = N)
  cal_ind = in_index3[1:m]
  tein_ind = in_index3[(m+1):N]
  tein_ind = sample(in_index3, size = n0)
  teout_ind = sample(out_ind, size = n1)
  cal = dataset[cal_ind,]
  te = dataset[c(tein_ind, teout_ind),]

  S_cal = predict.isolation_forest(iso.fo$model, cal, type = "score")
  S_te = predict.isolation_forest(iso.fo$model, te, type = "score")
  #U_i = sapply(1:n, function(i) sum(S_te[i]>S_cal))
  #U = sum(U_i)

  uniques[b] = length(unique(c(S_cal, S_te)))
  d_WMW[b] = nout::d_mannwhitney(S_Y=S_te, S_X=S_cal, crit = mycrit)
  #resU[b] = U >=mycrit$crit.vals[1]
  d_Simes[b] = nout::d_Simes(S_X=S_cal, S_Y=S_te, alpha=myalpha)
  d_StoSimes[b] = nout::d_StoreySimes(S_X=S_cal, S_Y=S_te, alpha=myalpha)$d
  d_BH[b] = nout::d_benjhoch(S_X=S_cal, S_Y=S_te, alpha=myalpha)
  d_StoBH[b] = nout::d_StoreyBH(S_X=S_cal, S_Y=S_te, alpha=myalpha)
}


discoveries = as.data.frame(cbind("d_BH"=d_BH, "d_StoBH"=d_StoBH, "d_Simes"=d_Simes,
                              "d_StoSimes"=d_StoSimes, "d_WMW"=d_WMW))
colnames(discoveries) = c("BH", "BHSto", "CTSim", "CTSimSto", "CTWMW")
mean.discoveries = apply(discoveries, MARGIN = 2, FUN = mean)

powerGlobalNull = as.data.frame(cbind("d_BH"=d_BH>0, "d_StoBH"=d_StoBH>0, "d_Simes"=d_Simes>0,
                                        "d_StoSimes"=d_StoSimes>0, "d_WMW"=d_WMW>0))
colnames(powerGlobalNull) = c("BH", "BHSto", "CTSim", "CTSimSto", "CTWMW")
mean.powerGlobalNull = apply(powerGlobalNull, MARGIN = 2, FUN = mean)

prova.d_WMW = mean(d_WMW>0)
#prova.resU = mean(resU)


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



```
mean.discoveries
```

```
##        BH    BHSto     CTSim  CTSimSto     CTWMW
##   32.6453  35.6630    8.5790    9.6293   30.7027
```

```
mean.powerGlobalNull
```

```
##        BH    BHSto    CTSim CTSimSto    CTWMW
##    0.9984   0.9988   0.9984   0.9988   0.9968
```

```
prova.d_WMW
```

```
## [1] 0.9968
```

```
#prova.resU

res=list("prova.d_WMW"=prova.d_WMW,
         #"prova.resU"=prova.resU,
         #"resU" = resU,
         "uniques"=uniques,
         "discoveries"=discoveries,
         "mean.discoveries" = mean.discoveries,
         "powerGlobalNull"=powerGlobalNull,
         "mean.powerGlobalNull"=mean.powerGlobalNull,
         "n1"=n1,
         "alpha"=myalpha)

resDigits10 = res
save(resDigits10,
     file="C:/Users/c.magnani9/Documents/nout/trials/RealData/PowerStudy/New&TidyNoSimuFunction/alpha0.
```