A Bootstrap Based Multiple Hypothesis Testing Procedure

Abhirup Mallik

November 26, 2024

We propose a new methodology for multiple hypothesis testing based on bootstrap distribution of p values. We know that under null hypothesis p value is an uniform$(0, 1)$ random variable. If the hypothesis are independent, then so are the p values. We use this intuition to construct a statistic that can be used to test multiple hypothesis.

We consider the case of coordinate wise testing of a data vector. If the data matrix X is of dimension $I \times J$ then without loss of generality, we consider the following set of hypothesis for $j = 1, 2, \cdots, J$.

$$H_0^{(j)} : x_j = 0$$
$$H_1^{(j)} : x_j \neq 0$$

Where x_j denotes the jth coordinate of X. This set of tests can be performed in a variety of ways, for the purpose of illustration, we choose to use one sample t-tests.

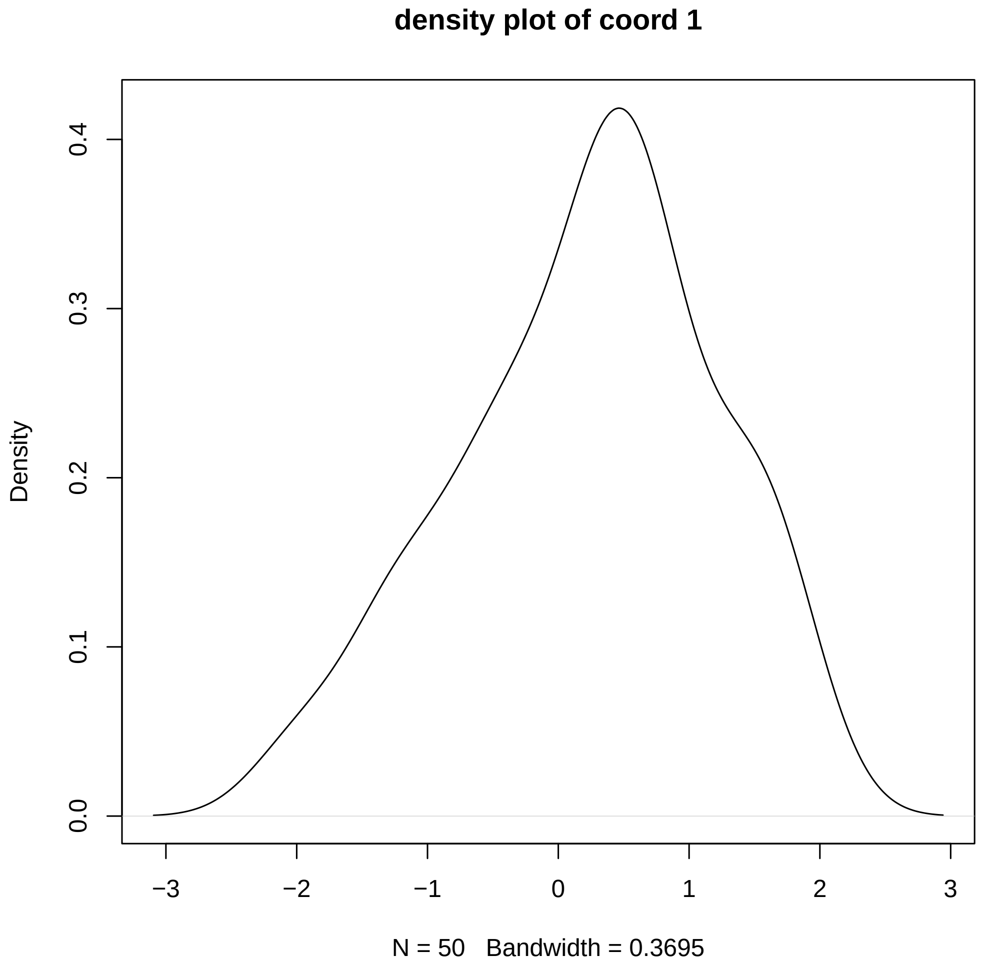
The general procedure for testing multiple hypothesis is testing independently then use a correction method to correct for family wise error rate. Here we are not using any such correction, as we are using a bootstrap based approach.

We first generate bootstrap samples of our data, let us denote the bootstrap samples by $X^{\{b\}}$ for $b \in \{1, 2, \cdots, B\}$, where B is the bootstrap sample size. For each of these samples we can perform J tests in parallel and collect the p values. We denote the p values from jth coordinate of bth bootstrap sample by $p_j^{\{b\}}$. We then use a monotone transformation of the p values for better visualizing. The transformed order statistics are collected as shown.

$$\begin{aligned} Z_{(0)}^{\{b\}} &\overset{\text{def}}{=} 0 \\ Z_{(j)}^{\{b\}} &= -\log(1 - p_j^{\{b\}}) \\ W_j^{\{b\}} &\overset{\text{def}}{=} Z_{(j)}^{\{b\}} - Z_{(j-1)}^{\{b\}} - (n+1-j)^{-1} \end{aligned}$$

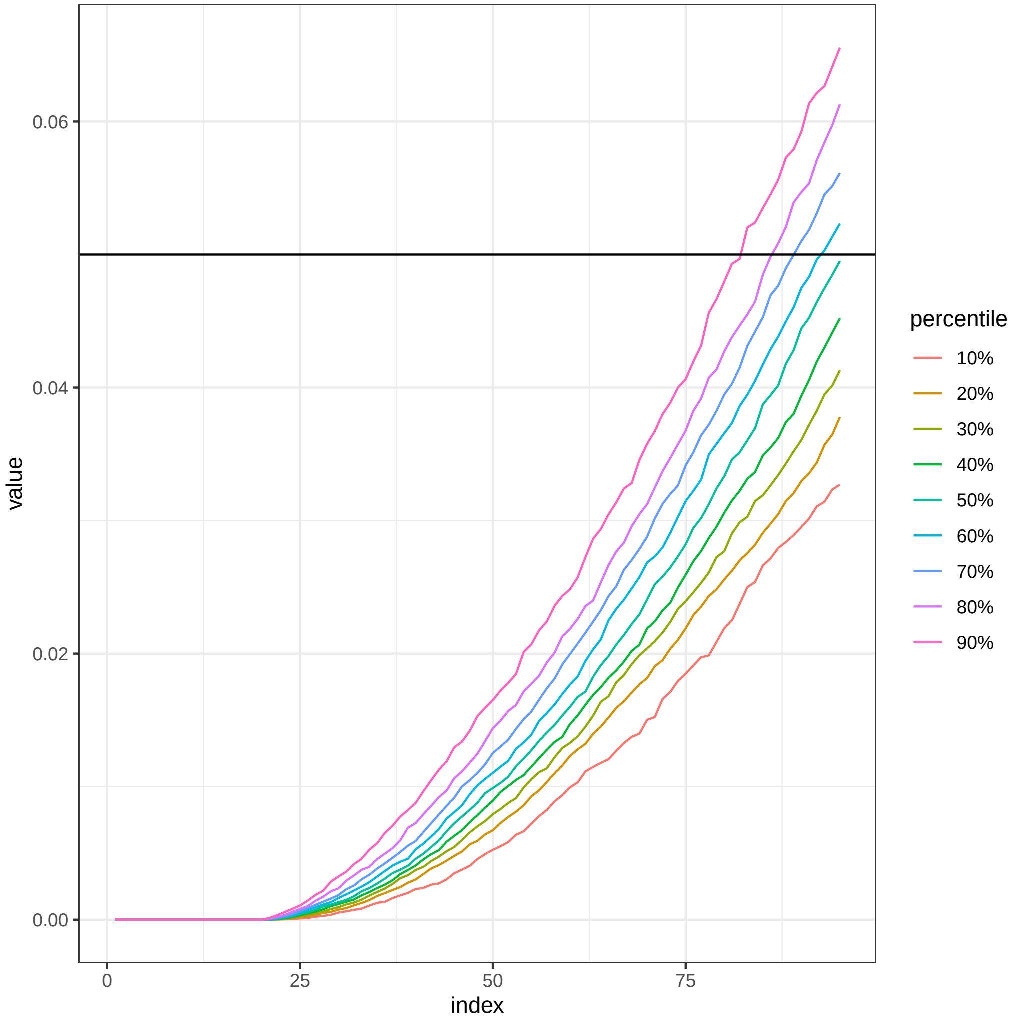
We demonstrate the proceedure below with a simulated example. The function datgen is used to simulate data from a multivariate normal distribution.

```
set.seed(12345)
suppressMessages(library(mhtboot))
n = 50;m = 500;m0 = 20;
sigeff = 1;
Sigma <- 0.25*diag(m)
X <- datgen(n,m,m0,sigeff,Sigma = Sigma)
plot(density(X[,1]),main="density plot of coord 1")
```



We then generate the distribution of the p values using bootstrap. This is impleemnted in the function series pboot. Here we are using the one sample version of the function. The plotpboot function is used to generate the quantile plot of the distribution of the p values.

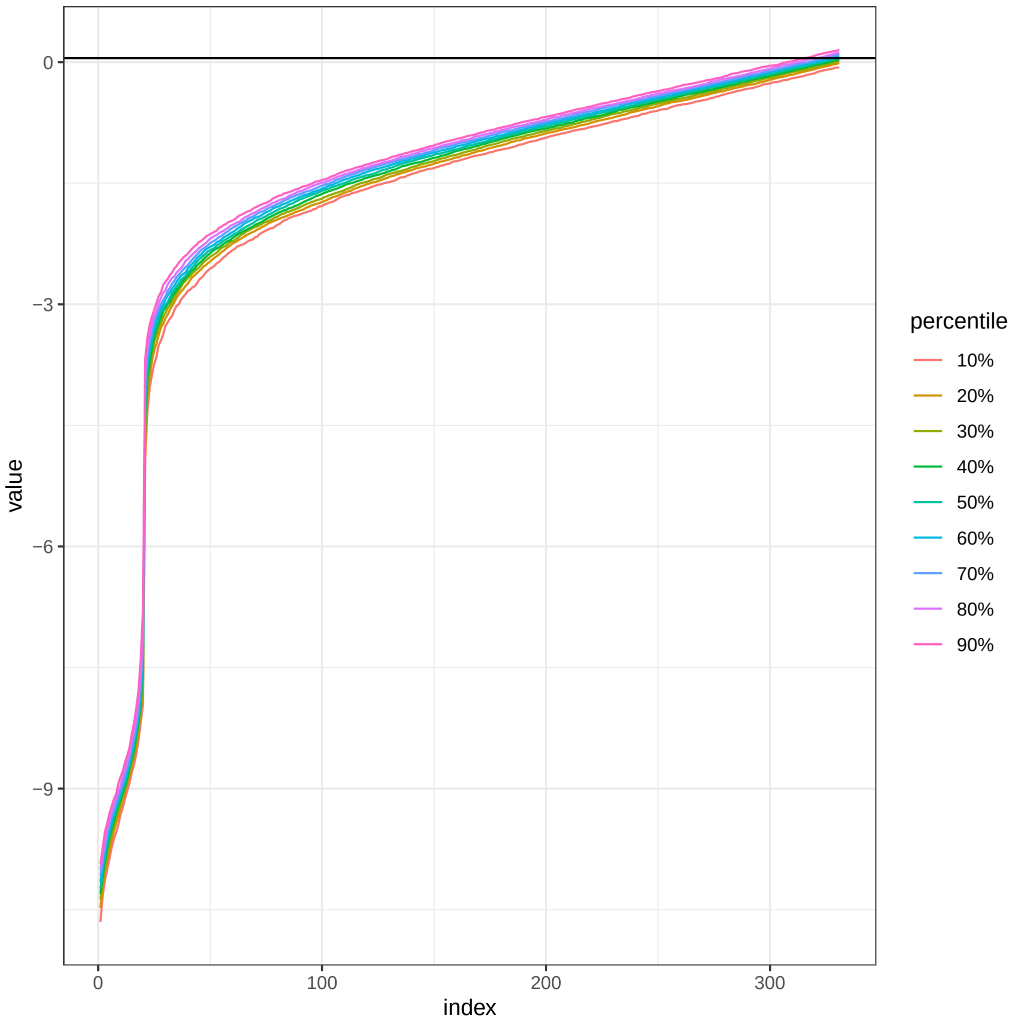
```
porder <- pboot.1sample(X=X,B=250,ncpus = 1)
plotpboot(porder = porder)
```



This approach can be extended to any set of hypothesis, in the package we also provide a function for two sample tests. Both of these functions can be used for user given tests as they accept test statistic as a parameter.

We can transform the order statistics of the p values using a monotone transformation. We show here the transformation using inverse normal cdf function.

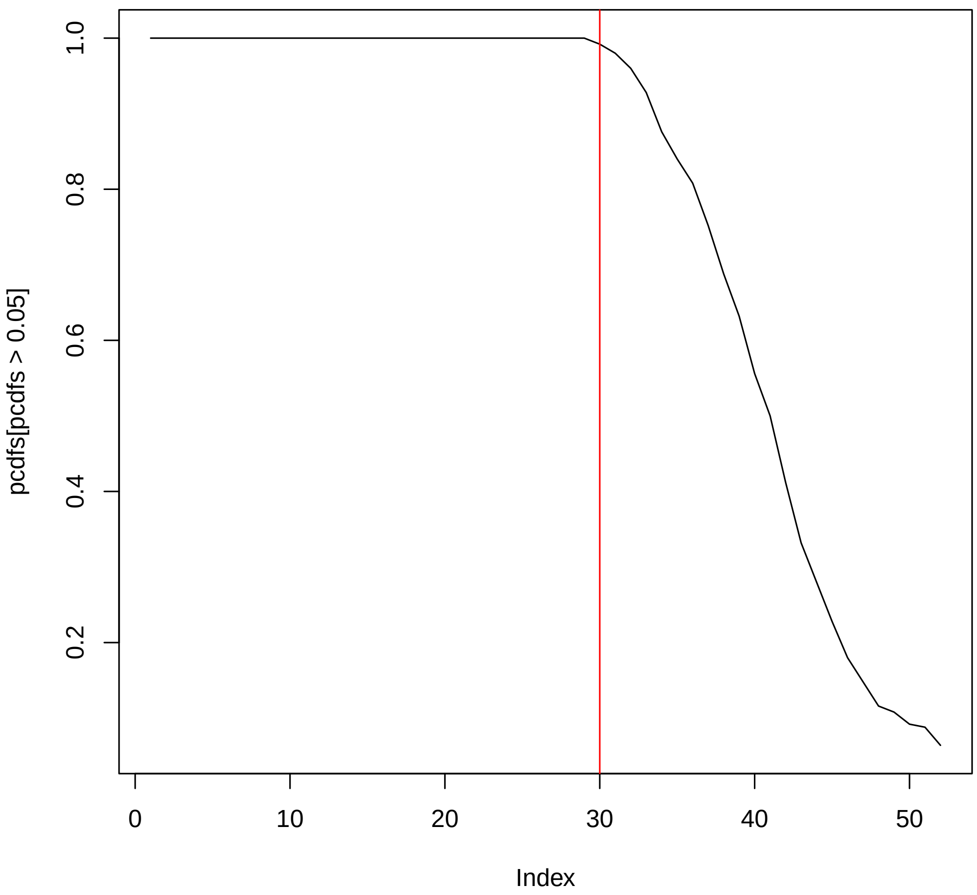
```
porder.tr <- ptrans(porder,trans="normal")
plotpboot(porder.tr)
```



We can also look at the points where each coordinates hit a certain probability.

```
porder <- ptrans(porder = porder)
hitplots(porder = porder,alpha = 0.005)

## [1] 30
```



Once we have the distribution of the p values, we can use them to detect the change point in their distribution. This is done through qelbow function.

```
out <- qelbow(porder = porder)
out

## dav dlm
##  32  21
```

All the above process is implemented in one single function for all one sample tests. The final cut off point is chosen by taking a minimum of two detection methods.

```
out1 <- mht.1sample(X,ncpus = 1)
out1$cutoff

## [1] 21

out1$signal

##  [1]  23  26  33  49  55  62  72  74 114 119 133 207 221 277 379 388 436 445 45
```