Package 'rcc'

October 14, 2022

Type Package

Title Parametric Bootstrapping to Control Rank Conditional Coverage

Version 1.0.0

Date 2017-02-21

Author Jean Morrison

Maintainer Jean Morrison <jeanm@uchicago.edu>

URL http://github.com/jean997/rcc

BugReports http://github.com/jean997/rcc/issues

Description Functions to implement the parametric and non-parametric bootstrap confidence interval methods described in Morrison and Simon (2017) <arXiv:1702.06986>.

Suggests parallel

License GPL (>= 2)

LazyData TRUE

NeedsCompilation no

RoxygenNote 6.0.1

Repository CRAN

Date/Publication 2017-02-28 23:38:34

R topics documented:

nonpar_bs_ci *Non-Parametric bootstrapped confidence intervals to control RCC*

Description

This function implements Algorithm 3 in the paper (see Section 2.4). The user supplies individual level data and an analysis function that generates test statistics and point estimates. The function resamples individuals from the original data and recalculates test statistics and estimates in order to calculate confidence intervals.

Usage

```
nonpar_bs_ci(data, analysis.func, rank.func = NULL, level = 0.9,
  res.orig = NULL, n.rep = 1000, use.abs = TRUE, parallel = FALSE, ...)
```

Arguments

data	An n by K matrix of data
analysis.func	A function that performs the analysis. This function should take exactly one argument (data) and return a list or data frame including an item called 'estimate' and an item called 'statistic'. These should both be vectors of length p, the number of parameters.
rank.func	A function that takes, as first argument, the statistics returned by analysis.func. The second argument should be use.abs which can take a logical value. This argument indicates if ranking should be based on signed or absolute value of the statistics. rank.func should return a list including items named order and rank. See rcc:::basic_rank for an example. If NULL, the basic_rank function will be used which ranks based on the size of the test statistics.
level	Confidence level
res.orig	Results of applying analysis.funct to the original data if they are already available. If NULL, these will be calculated.
n.rep	Number of bootstrap replications
use.abs	Logical. Rank based on absolute value of the statistics
parallel	Logical. If true, use the parallel package to make use of multiple cores.
...	Additional parameters to pass to rank.func

Value

A data frame giving original estimates and statistics, confidence intervals, debiased point estimates, and rank for each parameter.

Examples

```
#Generate some data -- 10 parameters, 30 samples
#Most problems will have many more parameters!
set.seed(4e8)
dat <- matrix(rnorm(n=10*30), nrow=10)

#Write a function to do a t-test comparing
#the first 15 samples and the last 15 samples
my.analysis.func <- function(data){
    x <- rep(c(1, 0), each=15)
    B <- t(apply(data, MARGIN=1, FUN=function(y){
        f <- t.test(y~x)
        est <- f$estimate[2]-f$estimate[1]
        stat <- f$statistic
        return(c(est, stat))
    }))
    B <- data.frame(B)
    names(B) <- c("estimate", "statistic")
    return(B)
}

#Calculate confidence intervals
cis <- nonpar_bs_ci(data=dat, analysis.func=my.analysis.func, n.rep=500)
head(cis)
```

---

par_bs_ci *Parametric bootstrapped confidence intervals to control RCC*

Description

This function implements the parametric bootstrap (see Section 2.3 of the referenced paper). The user supplies point estimates, standard errors and optionally, a ranking function.

Usage

```
par_bs_ci(beta, se = rep(1, length(beta)), rank.func = NULL, theta = beta,
  level = 0.9, n.rep = 1000, use.abs = TRUE, ...)
```

Arguments

beta	Parameter estimates
se	Estimated standard error of beta. Defaults to 1.
rank.func	A function that takes as first argument the t-statistics beta/se and returns a list with items order and rank. See rcc:::basic_rank for an example. If NULL, the basic_rank function will be used which ranks based on the size of the test statistics.
theta	Possibly shrunken estimates of E[beta]. Defaults to beta.

level	Confidence level
n.rep	Number of bootstrap replications
use.abs	Base the rank on abs(beta) rather than beta
...	Additional parameters to pass to rank.func

Value

A data frame giving original estimates and standard errors, confidence intervals, debiased point estimates, and rank for each parameter.

Examples

```
#generate 100 fake parameter estimates
theta <- c(rep(0, 90), rnorm(n=10)) #vector of means
beta <- rnorm(n=100, mean=theta, sd=1)
cis <- par_bs_ci(beta=beta, n.rep=500) #calculate parametric bootstrap confidence intervals
head(cis)
```

Index