Package: MChtest (via r-universe)

May 19, 2026

Type Package

Title Monte Carlo Hypothesis Tests with Sequential Stopping

Version 1.0-3

Date 2019-05-14

Author Michael P. Fay

Maintainer Michael P. Fay <mfay@niaid.nih.gov>

Depends stats

Description Performs Monte Carlo hypothesis tests, allowing a couple of different sequential stopping boundaries. For example, a truncated sequential probability ratio test boundary (Fay, Kim and Hachey, 2007 <DOI:10.1198/106186007X257025>) and a boundary proposed by Besag and Clifford, 1991 <DOI:10.1093/biomet/78.2.301>. Gives valid p-values and confidence intervals on p-values.

License GPL

URL https://www.niaid.nih.gov/about/brb-staff-fay

NeedsCompilation no

Repository https://michaelfay-niaid.r-universe.dev

Date/Publication 2019-05-16 12:10:03 UTC

RemoteUrl https://github.com/cran/MChtest

RemoteRef HEAD

RemoteSha 4b1c22dee02a6a6cb47e62ba67d5c6e548c2b034

Contents

`MChtest-package` *Monte Carlo hypothesis tests allowing sequential stopping*

Description

Performs Monte Carlo hypothesis tests. It allows a couple of different sequential stopping boundaries (a truncated sequential probability ratio test boundary and a boundary proposed by Besag and Clifford, 1991). Gives valid p-values and confidence intervals on p-values.

Details

Package:	MChtest
Type:	Package
Version:	1.0-3
Date:	2019-05-14
License:	GPL

Use `MCbound` to create sequential stopping boundaries. These may take considerable set-up time, but once the stopping boundary is calculated then it can be used in `MCtest` to save time in computation of Monte Carlo hypothesis tests. The idea of the truncated sequential probability ratio test boundary is that it takes many resamples if the true p-value (i.e., the one from an infinite resample size) is close to the significance level (e.g., 0.05), but takes much fewer if the true p-value is far from the significance level.

Author(s)

Michael P. Fay

Maintainer: Michael P. Fay <mfay@niaid.nih.gov>

References

Besag, J. and Clifford, P. (1991). Sequential Monte Carlo p-values. Biometrika. 78: 301-304.

Fay, M.P., Kim, H-J. and Hachey, M. (2007). Using truncated sequential probability ratio test boundaries for Monte Carlo implementation of hypothesis tests. Journal of Computational and Graphical Statistics. 16(4):946-967.

See Also

Precalculated MCbound: `MCbound.precalc1`

Examples

```
## Create a stopping boundary
##### May take a long time if Nmax is large
B<-MCbound("tsprt",c(alpha0=.001,beta0=.01,Nmax=99,p0=.04,p1=.06))
## do Monte Carlo  test
x<-data.frame(y=1:100,z=rnorm(100),group=c(rep(1,50),rep(2,50)))
stat<-function(x){ cor(x[,1],x[,2]) }
### nonparametric bootstrap test on correlation between y and z
### low p-value means that such a large correlation unlikely due to chance
resamp<-function(x){ n<-dim(x)[[1]] ; x[sample(1:n,replace=TRUE),] }
MCtest(x,stat,resamp,bound=B)
## Package comes with a large precalculated MC bound as the default
## the precalculated bound is good for testing at the 0.05 level
MCtest(x,stat,resamp)
```

---

find.ab *Find beta parameters to approximate distribution of p-values.*

Description

Find parameters of a beta distribution to approximate distribution of a p-value derived from a normal test statistic with one-sided significance level=ALPHA and power=1-BETA.

Usage

```
find.ab(n = 1e+05, ALPHA = 0.05, BETA = 0.2, higha = 100)
```

Arguments

n	the number of divisions for the numeric integration used to estimate the mean of p-value distribution, H (see details).
ALPHA	one-sided significance level of normal test statistic
BETA	type II error for normal test stastic
higha	an upper bound on the beta parameter (see details).

Details

The cumulative distribution function of the p-value from a normally distributed test statistic with one-sided significance level=ALPHA and power=1-BETA is H(p) = 1-pnorm(qnorm(1-p) - qnorm(1-ALPHA)+qnorm(BETA)). We approximate this distribution with a beta distribution, B, which has the same mean as H and has B(ALPHA)=1-BETA. If two beta distributions meet both those criteria, we select the one closest to H in terms of integrated square error of the cumulative distribution function. That error is estimated by the sample variance of the differences in the two CDFs evaluated at (0:n)/n. Note that the two beta distributions come from the two roots of the following function: 1-BETA - B(ALPHA) We search for those two roots as the beta parameter within the range (1/higha, higha).

Value

A list with two elements:

a	numeric value of one of the shape parameters of the beta distribution
b	numeric value of the other shape parameter of the beta distribution

Author(s)

M.P. Fay

References

Fay, M.P., and Follmann, D.A. (2002). "Designing Monte Carlo implementations of permutation or bootstrap hypothesis tests" American Statistician, 56: 63-70.

Examples

```
## See first line of Table 1, Fay and Follmann, 2002
find.ab(ALPHA=.05,BETA=.1)
```

MCbound *Create Monte Carlo stopping boundary*

Description

Creates one of several different types of Monte Carlo stopping boundaries

Usage

```
MCbound(type, parms, conf.level = 0.99)
```

Arguments

type	a character vector of type of boundary, possible values: "fixed", "tsprt","Bvalue", and "BC"
parms	a numeric vector of parameter values, different for each type (see details)
conf.level	confidence level for intervals about Monte Carlo p-values

Details

Create Monte Carlo stopping boundaries for use with MCtest, where we keep resampling until hitting the stopping boundary. There are several possible types, each with a different length parameter vector.

type="fixed"	then names(parms)=c("Nmax")
type="tsprt"	then names(parms)=c("p0","p1","A","B","Nmax")
	or names(parms)=c("p0","p1","alpha0","beta0","Nmax")

type="Bvalue"	then names(parms)=c("Nmax","alpha","e0","e1")
type="BC"	then names(parms)=c("Nmax","Smax")

The object parms should be a named vector, although unnamed vectors will work if the parameters are in the above order (for the tsprt it assumes the first parameterization). For type="fixed" we keep reampling until N=Nmax resamples. For type="tsprt" we keep resampling until stopping for a truncated sequential probability ratio test for a binary parmaeter. The parameterizations are the usual Wald notation, except alpha0=alpha and beta0=beta, where A=(1-beta0)/alpha0 and B=beta0/(1-alpha0). The Bvalue is a test that p=alpha or not and we stop if the B-value at information time t, B(t), is B(t)<= qnorm(e0) or B >=qnorm(1-e1). Note that the B-value stopping boundary is just a reparameterization of the truncated sequential probability ratio test. For type="BC" we keep resampling until N=Nmax or S=Smax following a design recommended by Besag and Clifford (1991). For each stopping boundary we calculate valid p-values at each stopping point ordering by S/N. For details see Fay, Kim and Hachey, 2006.

Value

An object of class MCbound. A list with the following elements:

`S`	number of sucesses at points on the boundary
`N`	number of resamples at points on the boundary
`p.value`	valid p-value at each point on boundary, calculated using ordering by S/N
`ci.lower`	lower confidence limit of p-value at each boundary point
`ci.upper`	upper confidence limit of p-value at each boundary point
`Kstar`	number of ways to reach each point, (S,N), on boundary times beta(S+1,N-S+1)
`conf.level`	confidence level for intervals on p-values
`type`	type of boundary: either "fixed", "tsprt", "Bvalue" or "BC"
`parms`	parameter vector that defines boundary (see details)

Author(s)

Michael P. Fay

References

Besag, J. and Clifford, P. (1991). Sequential Monte Carlo p-values. Biometrika. 78: 301-304.

Fay, M.P., Kim, H-J. and Hachey, M. (2007). Using truncated sequential probability ratio test boundaries for Monte Carlo implementation of hypothesis tests. Journal of Computational and Graphical Statistics. 16(4):946-967.

Examples

```
MCbound("tsprt",c(alpha0=.001,beta0=.01,Nmax=99,p0=.06,p1=.04))
```

`MCbound.precalc1` *Precalculated object of class MCbound*

Description

Because the calculation of this truncated sequential probability ratio test stopping boundary takes a long time, it is calculated ahead of time and included in the package. It is created with the following code, `MCbound("tsprt",parms=c(p0=p0.given.p1(0.04),p1=0.04,alpha0=.0001,beta0=0.0001,Nmax=9999),conf`

Usage

```
MCbound.precalc1
```

Format

The format is: List of 10

- S : num [1:10000] 22 22 22 22 22 22 22 22 22 22 ...
- N : num [1:10000] 22 23 24 25 26 27 28 29 30 31 ...
- p.value : num [1:10000] 1.000 0.957 0.917 0.880 0.846 ...
- ci.lower : num [1:10000] 0.786 0.719 0.668 0.626 0.590 ...
- ci.upper : num [1:10000] 1.000 1.000 0.995 0.985 0.972 ...
- Kstar : num [1:10000] 0.0435 0.0399 0.0367 0.0338 0.0313 ...
- conf.level: num 0.99
- type : chr "tsprt"
- parms : Named num [1:5] 6.14e-02 4.00e-02 1.00e-04 1.00e-04 1.00e+04
- ..- attr(*, "names")= chr [1:5] "p0" "p1" "alpha0" "beta0" ...
- check : num 1
- - attr(*, "class")= chr "MCbound"

Examples

```
plot(MCbound.precalc1)
```

`MCtest` *Perform Monte Carlo hypothesis tests.*

Description

Performs Monte Carlo hypothesis test with either a fixed number of resamples or a sequential stopping boundary on the number of resamples. Outputs p-value and confidence interval for p-value. The program is very general and different bootstrap or permutation tests may be done by defining the statistic function and the resample function.

Usage

```
MCtest(x, statistic, resample, bound, extreme = "geq", seed = 1234325)
MCtest.fixed(x, statistic, resample, Nmax, extreme = "geq",
    conf.level=.99, seed = 1234325)
```

Arguments

`x`	data object
`statistic`	function that inputs data object, x, and outputs a scalar numeric
`resample`	function that inputs data object,x, and outputs a "resampling" of x, an object of the same type as x
`Nmax`	the number of resamples for the fixed boundary
`bound`	a object of class MCbound, can be created by `MCbound` see that help
`extreme`	character value either "geq" or "leq". Defines which Monte Carlo outputs from statistic (T1,T2,...) are denoted extreme with respect to the original output (T0) ; extreme values are either all Ti >= T0 ("geq") or all Ti <= T0 ("leq").
`conf.level`	confidence level for interval about p-value
`seed`	a numeric value used in set.seed

Details

Performs Monte Carlo hypothesis test. `MCtest` allows any types of Monte Carlo boundary created by `MCbound`, while `MCtest.fixed` only performs Monte Carlo tests using fixed boundaries. The only advantage of `MCtest.fixed` is that one can do the test without first creating the fixed stopping boundary through `MCbound`. The default boundary is described in `MCbound.precalc1`.

Let T0=statistic(x) and let T1,T2,... be statistic(resample(x)). Then in the simplest type of boundary, the "fixed" type, then the resulting p-value is p=(S+1)/(N+1), where S=(\# Ti >= T0) (if extreme is "geq") or S=(\# Ti <= T0) (if extreme is "leq"). The confidence interval on the p-value is calculated by an exact method.

There are several different types of MC designs that may be used for the MCtest. These are described in the `MCbound` help.

Value

A LIST of class "MCtest",with elements:

Ti	an N vector of the outputs of the resampled statistic
type	type of boundary: "fixed", "tsprt", "Bvalue" or "BC"
parms	vector of parameters that define boundary specified by type
T0	the value of the test statistic applied to the original data
p.value	p.value
pvalue.ci	confidence interval about the p-value

Author(s)

M.P. Fay

References

Fay, M.P., Kim, H-J. and Hachey, M. (2007). Using truncated sequential probability ratio test boundaries for Monte Carlo implementation of hypothesis tests. Journal of Computational and Graphical Statistics. 16(4):946-967.

See Also

MCbound,MCbound.precalc1

Examples

```
x<-data.frame(y=1:100,z=rnorm(100),group=c(rep(1,50),rep(2,50)))
stat<-function(x){ cor(x[,1],x[,2]) }
### nonparametric bootstrap test on correlation between y and z
### low p-value means that such a large correlation unlikely due to chance
resamp<-function(x){ n<-dim(x)[[1]] ; x[sample(1:n,replace=TRUE),] }
out<-MCtest(x,stat,resamp,extreme="geq")
out$p.value
out$p.value.ci
### permutation test, permuting y only within group
resamp<-function(x){
    ug<-unique(x[,"group"])
    y<- x[,"y"]
    for (i in 1:length(ug)){
         pick.strata<- x[,"group"]==ug[i]
         y[pick.strata]<-sample(y[pick.strata],replace=FALSE)
    }
    x[,1]<-y
    x
}

out<-MCtest.fixed(x,stat,resamp,N=199)
out$p.value
out$p.value.ci
```

p1.given.p0 *Find p0 (or p1) associated with p1 (or p0) that gives minimax tsprt*

Description

Consider the SPRT for testing Ho:p=p0 vs H1:p=p1 where p1< Alpha < p0. For Monte Carlo tests, we want to reject and conclude that p<Alpha. In terms of the resampling risk at p (i.e., the probability of reaching a wrong decision at p) the minimax SPRT has a particular relationship between p0 and p1. Here we calculate p1 given p0 or vise versa to obtain that relationship.

Usage

```
p1.given.p0(p0, Alpha = 0.05, TOL = 10^-9)
p0.given.p1(p1, Alpha = 0.05, TOL = 10^-9)
```

Arguments

p0	null such that p0>Alpha
p1	alternative such that p1<Alpha
Alpha	want tsprt associated with testing p=Alpha or not
TOL	tolerance level input into call to uniroot

Value

either p0 or p1

Author(s)

Michael P. Fay

References

Fay, M.P., Kim, H-J. and Hachey, M. (2007). Using truncated sequential probability ratio test boundaries for Monte Carlo implementation of hypothesis tests. Journal of Computational and Graphical Statistics. 16(4):946-967.

Examples

```
p1.given.p0(.04)
```

`plot.MCbound` *Plot stopping boundary*

Description

Creates two plots of an object of class MCbound, an stopping boundary for use with Monte Carlo hypothesis tests. First, it plots the stopping boundary as number of replications (i.e., N) vs. number of sucesses (S). Second, it plots the estimated p-values vs. the confidence limits minus the estimated p-values (this nicely shows the width of the confidence intervals).

Usage

```
## S3 method for class 'MCbound'
plot(x, rdigit=4, plimit=500,...)
```

Arguments

`x`	an object of class MCbound
`rdigit`	the rounding digit for the parms values in the title
`plimit`	if the number of points in the MCbound is >plimit then plot lines, otherwise plot points
`...`	additional arguments to both plot functions

Value

Does not return any values. Does two plots only.

Author(s)

M.P. Fay

See Also

MCbound

Examples

```
plot(MCbound.precalc1)
```

rrisk *Calculate resampling risk and expected resampling size*

Description

Calculates for a particular stopping boundary the resampling risk of making the wrong accept/reject decision. Can be calculated for different distributions of the p-value. If type="p" then assume point mass at pparms. If type="b" then assume a beta distribution with two shape parameters given by pparms.

Usage

```
rrisk(bound, pparms, sig.level = 0.05, type = "b")
```

Arguments

bound	an object of class MCbound, i.e., a stopping boundary. See MCbound to create
pparms	either a vector of possible point mass p-value distributions (type="p"), or a vector (or matrix with two columns) representing two beta shape parameters (type="b")
sig.level	significance level for defining resampling risk
type	either "p" for point mass p-value distributions, or "b" for a beta distribution

Details

The resampling risk (RR) is defined as the probability of making an accept/reject decision different from complete enumeration. In other words, for any Monte Carlo test the true p-value for any data is either below the sig.level (reject the null) or above the sig.level (accept the null), and the RR is the probability of either deciding p<=sig.level when p>sig.level or vise versa. We also calculate the expected resampling size for the assumed distributions on the p-values. As a check of the MCbound, we sum the probability of stopping at any point in the boundary over the entire stopping boundary for each assumed distribution on the p-values; the ouput value check should give a vector of all ones if the MCbound is calculated correctly.

Value

A list with the following elements:

check	Sum of the probabilities of the stopping boundary corresponding to the p-value distribution(s). Should be a vector with all values equal to 1.
rr	resampling risk corresponding to the p-value distribution(s)
EN	expected resampling size corresponding to the p-value distribution(s)

Author(s)

Michael P. Fay

References

Fay, M.P., Kim, H-J. and Hachey, M. (2007). Using truncated sequential probability ratio test boundaries for Monte Carlo implementation of hypothesis tests. Journal of Computational and Graphical Statistics. 16(4):946-967.

Examples

```
### caculate resampling risk and E(N) under null, i.e., uniform distribution on p-values
rrisk(MCbound.precalc1,c(1,1))
```

`tSPRT.to.Bvalue` *Convert between MCbound parameterizations*

Description

Convert from the tSPRT to the Bvalue parametrization or vice versa.

Usage

```
tSPRT.to.Bvalue(parms)
Bvalue.to.tSPRT(parms,p0,TOL=10^-8)
```

Arguments

`parms`	named vector of parameters
`p0`	To pick a unique parameterization of the type tSPRT, you must specify p0
`TOL`	tolerance for solution of `p1.given.p0`

Value

Parameter vector of other parameterization.

Note

`tsprt.to.Bvalue` called by `MCbound` when type="tsprt".

Author(s)

Michael P. Fay

See Also

`MCbound`

Examples

```
temp<-tSPRT.to.Bvalue(c(p0=.04,p1=p1.given.p0(.04),alpha0=.001,beta0=.001,Nmax=9999))
temp
Bvalue.to.tSPRT(temp,p0=.04)
```

Index