Package ‘mmcm’

February 26, 2018

Type Package
Title Modified Maximum Contrast Method
Version 1.2-7
Imports mvtnorm, OpenMPController
License GPL-3
RoxygenNote 6.0.1
NeedsCompilation yes
Author Kengo Nagashima [aut, cre] (<https://orcid.org/0000-0003-4529-9045>), Yasunori Sato [aut]
Maintainer Kengo Nagashima <nsh1201@gmail.com>
Repository CRAN
Date/Publication 2018-02-26 11:42:42 UTC

R topics documented:

  mmcm-package ........................................................... 2
  mcm.mvt .............................................................. 2
  mmcm.mvt ............................................................ 5
  mmcm.resamp ......................................................... 8
  print.mmcm .......................................................... 12

Index 13
Description

This package provides an implementation of modified maximum contrast methods and the maximum contrast method. This version supports functions `mmcm.mvt`, `mcm.mvt` that gives P-value by using randomized quasi-Monte Carlo method from `pmvt` function of package `mvtnorm`, and `mmcm.resamp` that gives P-value by using the permutation method. In a one-way problem testing pattern of several factor level means, the maximum contrast statistics (Yoshimura, I., 1997) may be used. But under unequal sample size situations, denominator of the maximum contrast statistics is overestimated. Thus we propose a modified maximum contrast statistics for the unequal sample size situation. Denominator of the modified maximum contrast statistics is not influenced under the unequal sample size situation.

References


See Also

`mcm.mvt`, `mmcm.mvt`, `mmcm.resamp`

mcm.mvt

The maximum contrast method by using the randomized quasi-Monte Carlo method

Description

This function gives P-value for the maximum contrast statistics by using randomized quasi-Monte Carlo method from `pmvt` function of package `mvtnorm`.

Usage

```r
mcm.mvt(x, g, contrast, alternative = c("two.sided", "less", "greater"),
algorithm = GenzBretz())
```
Arguments

- **x** a numeric vector of data values
- **g** an integer vector giving the group for the corresponding elements of x
- **contrast** a numeric contrast coefficient matrix for the maximum contrast statistics
- **alternative** a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". You can specify just the initial letter.
- **algorithm** an object of class `GenzBretz` defining the hyper parameters of this algorithm

Details

`mcm.mvt` performs the maximum contrast method that is detecting a true response pattern.

Y_{ij}(i = 1, 2, \ldots; j = 1, 2, \ldots, n_i) is an observed response for j-th individual in i-th group.

C is coefficient matrix for the maximum contrast statistics (i × k matrix, i: No. of groups, k: No. of pattern).

\[ C = (c_1, c_2, \ldots, c_k)^T \]

c_k is coefficient vector of kth pattern.

\[ c_k = (c_{k1}, c_{k2}, \ldots, c_{ki})^T \quad (\sum_i c_{ki} = 0) \]

T_{max} is the maximum contrast statistic.

\[ \bar{Y}_i = \frac{\sum_{j=1}^{n_i} Y_{ij}}{n_i}, \bar{Y} = (\bar{Y}_1, \bar{Y}_2, \ldots, \bar{Y}_i, \ldots, \bar{Y}_a)^T, \]

\[ D = \text{diag}(n_1, n_2, \ldots, n_i, \ldots, n_a), V = \frac{1}{\gamma} \sum_{j=1}^{n_i} \sum_{i=1}^{a} (Y_{ij} - \bar{Y}_i)^2, \]

\[ \gamma = \sum_{i=1}^{a} (n_i - 1), T_k = \frac{c_k^T \bar{Y}}{\sqrt{Vc_k^T Dc_k}}, \]

\[ T_{\text{max}} = \max(T_1, T_2, \ldots, T_k). \]

Consider testing the overall null hypothesis H_0 : \mu_1 = \mu_2 = \ldots = \mu_i, versus alternative hypotheses H_1 for response patterns (H_1 : \mu_1 < \mu_2 < \ldots < \mu_i, \mu_1 = \mu_2 < \ldots < \mu_i, \mu_1 < \mu_2 < \ldots = \mu_i). The P-value for the probability distribution of T_{max} under the overall null hypothesis is

\[ P\text{-value} = \Pr(T_{\text{max}} > t_{\text{max}} \mid H_0) \]

T_{\text{max}} is observed value of statistics. This function gives distribution of T_{max} by using randomized quasi-Monte Carlo method from package `mvtnorm`.

\[ mcm.mvt \]
Value

- statistic: the value of the test statistic with a name describing it.
- p.value: the p-value for the test.
- alternative: a character string describing the alternative hypothesis.
- method: the type of test applied.
- contrast: a character string giving the names of the data.
- contrast.index: a suffix of coefficient vector of the kth pattern that gives maximum contrast statistics (row number of the coefficient matrix).
- error: estimated absolute error and,
- msg: status messages.

References


See Also

pmvt, GenzBretz, mmcm.mvt

Examples

```r
## Example 1 ##
# true response pattern: dominant model c=(1, 1, -2)
set.seed(136885)
x <- c(
  rnorm(130, mean = 1 / 6, sd = 1),
  rnorm( 90, mean = 1 / 6, sd = 1),
  rnorm( 10, mean = -2 / 6, sd = 1)
)
g <- rep(1:3, c(130, 90, 10))
boxplot(
  x ~ g,
  width = c(length(g==1), length(g==2), length(g==3)),
  main = "Dominant model (sample data)",
  xlab = "Genotype",
  ylab = "PK parameter"
)

# coefficient matrix
# c_1: additive, c_2: recessive, c_3: dominant
contrast <- rbind(
  c(-1, 0, 1), c(-2, 1, 1), c(-1, -1, 2)
)
y <- mcm.mvt(x, g, contrast)
y
## Example 2 ##
# for dataframe
```
The modified maximum contrast method by using randomized quasi-Monte Carlo method

Description

This function gives $P$-value for the modified maximum contrast statistics by using randomized quasi-Monte Carlo method from `pmvtnorm` function of package `mvtnorm`.

```r
# true response pattern: pos = 1 dominant model c=( 1, 1, -2)
# 2 additive model c=(-1, 0, 1)
# 3 recessive model c=( 2, -1, -1)
set.seed(3872435)
x <- c(
  rnorm(130, mean = 1 / 6, sd = 1),
  rnorm( 90, mean = 1 / 6, sd = 1),
  rnorm(10, mean = -2 / 6, sd = 1),
  rnorm(130, mean = -1 / 4, sd = 1),
  rnorm( 90, mean = 0 / 4, sd = 1),
  rnorm(10, mean = 1 / 4, sd = 1),
  rnorm(130, mean = 2 / 6, sd = 1),
  rnorm( 90, mean = -1 / 6, sd = 1),
  rnorm(10, mean = -1 / 6, sd = 1)
)
g <- rep(rep(1:3, c(130, 90, 10)), 3)
pos <- rep(c("rsXXXX", "rsYYYY", "rsZZZZ"), each=230)
xx <- data.frame(pos = pos, x = x, g = g)

# coefficient matrix
# c_1: additive, c_2: recessive, c_3: dominant
contrast <- rbind(
  c(-1, 0, 1), c(-2, 1, 1), c(-1, -1, 2)
)
mmcmtapply <- function(r) {
  mcm.mvt(
    xx$x[xx$pos==r[1]], xx$g[xx$pos==r[1]],
    contrast
  )
}
y <- tapply(xx$pos, xx$pos, mmcmtapply)

yy <- data.frame(
  Pos = as.vector(names(y)),
  Pval = as.vector(sapply(y, "[", 3)),
  Pattern = as.vector(sapply(y, "[", 7)),
  QMC_Error = as.vector(sapply(y, "[", 9))
)
# miss-detection!

```
Usage

\[ \text{mmcm.mvt}(x, g, \text{contrast}, \text{alternative} = \text{c("two.sided"}, "less", "greater"), \\
\quad \text{algorithm} = \text{GenzBretz()}) \]

Arguments

- \( x \) a numeric vector of data values
- \( g \) a integer vector giving the group for the corresponding elements of \( x \)
- \( \text{contrast} \) a numeric contrast coefficient matrix for modified maximum contrast statistics
- \( \text{alternative} \) a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". You can specify just the initial letter.
- \( \text{algorithm} \) an object of class \text{GenzBretz} defining the hyper parameters of this algorithm.

Details

\text{mmcm.mvt} performs the modified maximum contrast method that is detecting a true response pattern under the unequal sample size situation.

\( Y_{ij} (i = 1, 2, \ldots; j = 1, 2, \ldots, n_i) \) is an observed response for \( j \)-th individual in \( i \)-th group.

\( C \) is coefficient matrix for modified maximum contrast statistics (\( i \times k \) matrix, \( i \): No. of groups, \( k \): No. of pattern).

\[ C = (c_1, c_2, \ldots, c_k)^T \]

\( c_k \) is coefficient vector of \( k \)th pattern.

\[ c_k = (c_{k1}, c_{k2}, \ldots, c_{ki})^T \quad (\sum_i c_{ki} = 0) \]

\( S_{\text{max}} \) is the modified maximum contrast statistic.

\[ \bar{Y}_i = \frac{\sum_{j=1}^{n_i} Y_{ij}}{n_i}, \bar{Y} = (\bar{Y}_1, \bar{Y}_2, \ldots, \bar{Y}_i, \ldots, \bar{Y}_a)^T, \]

\[ V = \frac{1}{\gamma} \sum_{j=1}^{n_i} \sum_{i=1}^{a} (Y_{ij} - \bar{Y}_i)^2, \gamma = \sum_{i=1}^{a} (n_i - 1), \]

\[ S_k = \frac{c_i^T \bar{Y}}{\sqrt{Vc_i^Tc_k}} \]

\[ S_{\text{max}} = \max(S_1, S_2, \ldots, S_k). \]

Consider testing the overall null hypothesis \( H_0 : \mu_1 = \mu_2 = \ldots = \mu_i \), versus alternative hypotheses \( H_1 \) for response patterns \( (H_1 : \mu_1 < \mu_2 < \ldots < \mu_i, \mu_1 = \mu_2 < \ldots < \mu_i, \mu_1 < \mu_2 < \ldots = \mu_i) \). The \( P \)-value for the probability distribution of \( S_{\text{max}} \) under the overall null hypothesis is

\[ P\text{-value} = \Pr(S_{\text{max}} > s_{\text{max}} \mid H_0) \]

\( s_{\text{max}} \) is observed value of statistics. This function gives distribution of \( S_{\text{max}} \) by using randomized quasi-Monte Carlo method from package \text{mvtnorm}.
Value

statistic  the value of the test statistic with a name describing it.
p.value   the p-value for the test.
am水准ite  a character string describing the alternative hypothesis.
method    the type of test applied.
contrast  a character string giving the names of the data.
contrast.index a suffix of coefficient vector of the \( k \)th pattern that gives modified maximum contrast statistics (row number of the coefficient matrix).
error     estimated absolute error and,
msg       status messages.

References


See Also

pmvt, GenzBretz, mmcm.resamp

Examples

### Example 1 ###

# true response pattern: dominant model c=(1, 1, -2)
set.seed(136885)
x <- c(
    rnorm(130, mean = 1/6, sd = 1),
    rnorm( 90, mean = 1/6, sd = 1),
    rnorm( 10, mean = -2/6, sd = 1)
)
g <- rep(1:3, c(130, 90, 10))
boxplot(
    x ~ g,
    width = c(length(g[g==1]), length(g[g==2]), length(g[g==3])),
    main = "Dominant model (sample data)",
    xlab = "Genotype", ylab="PK parameter"
)

# coefficient matrix
# c_1: additive, c_2: recessive, c_3: dominant
contrast <- rbind(
    c(-1, 0, 1), c(-2, 1, 1), c(-1, -1, 2)
)
y <- mmcm.mvt(x, g, contrast)
The permuted modified maximum contrast method

**Description**

This function gives \( P \)-value for the permuted modified maximum contrast method.
Usage

mmcm.resamp(x, g, contrast, alternative = c("two.sided", "less", "greater"), nsample = 20000, abseps = 0.001, seed = NULL)

Arguments

x  a numeric vector of data values

G  a integer vector giving the group for the corresponding elements of x

contrast  a numeric contrast coefficient matrix for permuted modified maximum contrast statistics

alternative  a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". You can specify just the initial letter.

nsample  specifies the number of resamples (default: 20000)

abseps  specifies the absolute error tolerance (default: 0.001)

seed  a single value, interpreted as an integer; see set.seed() function. (default: NULL)

Details

mmcm.resamp performs the permutated modified maximum contrast method that is detecting a true response pattern under the unequal sample size situation.

\( Y_{ij} (i = 1, 2, \ldots; j = 1, 2, \ldots, n_i) \) is an observed response for \( j \)-th individual in \( i \)-th group.

\( C \) is coefficient matrix for permuted modified maximum contrast statistics (\( i \times k \) matrix, \( i \): No. of groups, \( k \): No. of pattern).

\[ C = (c_1, c_2, \ldots, c_k)^T \]

\( c_k \) is coefficient vector of \( k \)-th pattern.

\[ c_k = (c_{k1}, c_{k2}, \ldots, c_{ki})^T \quad (\sum c_{ki} = 0) \]

\( M_{\text{max}} \) is a permutated modified maximum contrast statistic.

\[ \bar{Y}_i = \frac{\sum_{j=1}^{n_i} Y_{ij}}{n_i}, \quad \bar{Y} = (\bar{Y}_1, \bar{Y}_2, \ldots, \bar{Y}_i, \ldots, \bar{Y}_a)^T, \quad M_k = \frac{c_k^T \bar{Y}}{\sqrt{c_k^T c_k}} \]

\[ M_{\text{max}} = \max(M_1, M_2, \ldots, M_k). \]

Consider testing the overall null hypothesis \( H_0 : \mu_1 = \mu_2 = \ldots = \mu_i \), versus alternative hypotheses \( H_1 \) for response patterns (\( H_1 : \mu_1 < \mu_2 < \ldots < \mu_i, \mu_1 = \mu_2 < \ldots < \mu_i, \mu_1 < \mu_2 < \ldots = \mu_i \)). The \( P \)-value for the probability distribution of \( M_{\text{max}} \) under the overall null hypothesis is

\[ P\text{-value} = \Pr(M_{\text{max}} > m_{\text{max}} \mid H_0) \]

\( m_{\text{max}} \) is observed value of statistics. This function gives distribution of \( M_{\text{max}} \) by using the permutation method, follow algorithm:

1. Initialize counting variable: \( COUNT = 0 \). Input parameters: \( NRESAMP\text{MIN} \) (minimum resampling count, we set 1000), \( NRESAMP\text{MAX} \) (maximum resampling count), and \( \epsilon \) (absolute error tolerance).
2. Calculate $m_{\text{max}}$ that is the observed value of the test statistic.

3. Let $y_{ij}^{(r)}$ donate data, which are sampled without replacement, and independently, form observed value $y_{ij}$. Where, $(r)$ is suffix of the resampling number ($r = 1, 2, \ldots$).

4. Calculate $m_{\text{max}}^{(r)}$ from $y_{ij}^{(r)}$. If $m_{\text{max}}^{(r)} > m_{\text{max}}$, then increment the counting variable: $\text{COUNT} = \text{COUNT} + 1$. Calculate approximate P-value $\hat{p}^{(r)} = \text{COUNT} / r$, and the simulation standard error $SE(\hat{p}^{(r)}) = \sqrt{\hat{p}^{(r)}(1 - \hat{p}^{(r)})} / r$.

5. Repeat 3–4, while $r > 1000$ and $3.5SE(\hat{p}^{(r)}) < \epsilon$ (corresponding to 99% confidence level), or $N_{\text{RESAMP MAX}}$ times. Output the approximate P-value $\hat{p}^{(r)}$.

Value

- **statistic**: the value of the test statistic with a name describing it.
- **p.value**: the p-value for the test.
- **alternative**: a character string describing the alternative hypothesis.
- **method**: the type of test applied.
- **contrast**: a character string giving the names of the data.
- **contrast.index**: a suffix of coefficient vector of the $k$th pattern that gives permuted modified maximum contrast statistics (row number of the coefficient matrix).
- **error**: estimated absolute error and,
- **msg**: status messages.

References


See Also

- `mmcm.mvt`

Examples

```r
OpenMPController::omp_set_num_threads(2)

## Example 1 ##
# true response pattern: dominant model c=(1, 1, -2)
set.seed(136885)
x <- c(
  rnorm(130, mean = 1 / 6, sd = 1),
  rnorm( 90, mean = 1 / 6, sd = 1),
  rnorm( 10, mean = -2 / 6, sd = 1)
)
```
```r
# coefficient matrix
# c_1: additive, c_2: recessive, c_3: dominant
contrast <- rbind(
  c(-1, 0, 1), c(-2, 1, 1), c(-1, -1, 2)
)
y <- mmcm.resamp(x, g, contrast, nsample = 20000, abseps = 0.01, seed = 5784324)

## Example 2 ##
# for dataframe
# true response pattern: pos = 1 dominant model c=( 1, 1, -2)
#                      2 additive model c=(-1, 0, 1)
#                      3 recessive model c=( 2, -1, -1)
set.seed(3872435)
x <- c(
  rnorm(130, mean = 1 / 6, sd = 1),
  rnorm( 90, mean = 1 / 6, sd = 1),
  rnorm( 10, mean = -2 / 6, sd = 1),
  rnorm(130, mean = -1 / 4, sd = 1),
  rnorm( 90, mean =  0 / 4, sd = 1),
  rnorm( 10, mean =  1 / 4, sd = 1),
  rnorm(130, mean =  2 / 6, sd = 1),
  rnorm( 90, mean = -1 / 6, sd = 1),
  rnorm( 10, mean = -1 / 6, sd = 1)
)
g <- rep(rep(1:3, c(130, 90, 10)), 3)
pos <- rep(c("rsXXXX", "rsYYYY", "rsZZZZ"), each=230)
xx <- data.frame(pos = pos, x = x, g = g)

# coefficient matrix
# c_1: additive, c_2: recessive, c_3: dominant
contrast <- rbind(
  c(-1, 0, 1), c(-2, 1, 1), c(-1, -1, 2)
)

mmcmapply <- function(r) {
  mmcm.resamp(
    xx$x[xx$pos==r[1]], xx$g[xx$pos==r[1]],
    contrast, nsample = 10000, abseps = 0.01, seed = 5784324+as.numeric(r[1])
  )
}
y <- tapply(xx$pos, xx$pos, mmcmapply)
yy <- data.frame(  
  Pos     = as.vector(names(y)),  
  Pval    = as.vector(sapply(y, "[", 3)))
```
Pattern = as.vector(sapply(y, "[[", ?)),
MC_Error = as.vector(sapply(y, "[[", 9))
)

print.mmcm

---

**print.mmcm**

*Print function for mmcm object*

**Description**

This function print result of function `mcm.mvt`, `mmcm.mvt` and `mmcm.resamp`

**Usage**

```r
## S3 method for class 'mmcm'
print(x, digits = getOption("digits"), ...)
```

**Arguments**

- `x`: Object of class `mmcm`, which is result of function `mcm.mvt`, `mmcm.mvt` and `mmcm.resamp`.
- `digits`: a non-null value for digits specifies the minimum number of significant digits to be printed in values. The default, NULL, uses `getOption(digits)`. (For the interpretation for complex numbers see `signif`.) Non-integer values will be rounded down, and only values greater than or equal to 1 and no greater than 22 are accepted.
- `...`: Further arguments passed to or from other methods.

**Details**

The case where printed "More than 2 contrast coefficient vectors were selected", some contrast may be unsuitable.

**See Also**

`print.default`, `mmcm.mvt`, `mmcm.resamp`, `mcm.mvt`
Index

*Topic **htest**
  mcm.mvt, 2
  mmcm-package, 2
  mmcm.mvt, 5
  mmcm.resamp, 8

*Topic **print**
  print.mmc, 12

GenzBretz, 3, 4, 6, 7
getOption, 12

mcm.mvt, 2, 2, 3, 12
mmcm (mmcm-package), 2
mmcm-package, 2
mmcm.mvt, 2, 4, 5, 6, 10, 12
mmcm.resamp, 2, 7, 8, 9, 12

pmvt, 2, 4, 5, 7
print.default, 12
print.mmc, 12

set.seed(), 9
signif, 12