Package ‘MDMR’

October 12, 2022

Type Package
Title Multivariate Distance Matrix Regression
Version 0.5.1
Date 2018-02-11
Author Daniel B. McArtor (dmcartor@gmail.com) [aut, cre]
Maintainer Daniel B. McArtor <dmcartor@gmail.com>
License GPL (>= 2)
LazyData TRUE
Depends R (>= 3.0.0),
Imports CompQuadForm, parallel, lme4, car
Suggests knitr, rmarkdown
VignetteBuilder knitr
URL http://github.com/dmcartor/mdmr
BugReports http://github.com/dmcartor/mdmr/issues
RoxygenNote 6.0.1
NeedsCompilation no
Repository CRAN
Date/Publication 2018-02-14 18:16:13 UTC

R topics documented:

  MDMR-package ......................................................... 2
delta ................................................................. 3
gower .............................................................. 6
mdmr ................................................................. 6
mixed.mdmr .......................................................... 9
**MDMR-package**

**Description**

MDMR allows a user to conduct multivariate distance matrix regression using analytic p-values and measures of effect size described by McArtor et al. (2017). Analytic p-values are computed using the R package CompQuadForm (Duchesne & De Micheaux, 2010). It also facilitates the use of MDMR on samples consisting of (hierarchically) clustered observations.

**Usage**

To access this package’s tutorial, type the following line into the console:

```r
vignette('mdmr-vignette')
```

There are three primary functions that comprise this package: `mdmr`, which regresses a distance matrix onto a set of predictors, and `delta`, which computes measures of univariate effect size in the context of multivariate distance matrix regression. The third function `mixed.mdmr` facilitates the use of MDMR on (hierarchically) clustered samples using an approach analogous to the linearar mixed-effects model for univariate outcomes. The help files of all three functions provide more general information than the package vignette.

**References**


**Examples**

```
# Conducting MDMR on data comprised of independent observations
# Source data
data(mdmrdata)

# Get distance matrix
D <- dist(Y.mdmr, method = 'euclidean')

# Conduct MDMR
mdmr.res <- mdmr(X = X.mdmr, D = D)
summary(mdmr.res)
```

```
# Conducting MDMR on data comprised of dependent observations
# Source data
data("clustmdmrdata")

# Get distance matrix
D <- dist(Y.clust)

# Conduct mixed-MDMR
mixed.res <- mixed.mdmr(~ x1 + x2 + (x1 + x2 | grp),
                        data = X.clust, D = D)
summary(mixed.res)
```

---

**delta**  
*Compute univariate MDMR effect sizes*

**Description**

`delta` computes permutation-based effect sizes on individual items comprising the distance matrix outcome used in multivariate distance matrix regression. It returns the omnibus estimates of delta (i.e. effect size of the entire design matrix on each outcome) as well as estimates of each pair-wise effect size (i.e. the effect of each predictor on each outcome variable, conditional on the rest of the predictors).

**Usage**

```
delta(X, Y = NULL, dtype = NULL, niter = 10, x.inds = NULL,
      y.inds = NULL, G = NULL, G.list = NULL, ncores = 1, seed = NULL,
      plot.res = F, grayscale = F, cex = 1, y.las = 2)
```
Arguments

X  A \(n \times p\) matrix or data frame of predictors. Unordered factors will be tested with contrast-codes by default, and ordered factors will be tested with polynomial contrasts. For finer control of how categorical predictors are handled, or if higher-order effects are desired, the output from a call to `model.matrix()` can be supplied to this argument as well.

Y  Outcome data: \(n \times q\) matrix of scores along the dependent variables.

dtype  Measure of dissimilarity that will be used by `dist` to compute the distance matrix based on \(Y\). As is the case when calling `dist` directly, this must be one of “euclidean”, “maximum”, “manhattan”, “canberra”, “binary” or “minkowski”, and unambiguous substring can be given.

niter  Number of times to permute each outcome item in the procedure to compute delta. The final result is the average of all `niter` iterations. Higher values of `niter` require more computation time, but result in more precise estimates.

x.inds  Vector indicating which columns of \(X\) should have their conditional effect sizes computed. Default value of `NULL` results in all effects being computed, and a value of 0 results in no conditional effects being computed, such that only the omnibus effect sizes will be reported.

y.inds  Vector indicating which columns of \(Y\) effect sizes should be computed on. Default value of `NULL` results in all columns being used.

G  Gower's centered similarity matrix computed from \(D\). Either \(D\) or \(G\) must be passed to `mdmr()`.

G.list  List of length \(q\) where the \(i^{th}\) element contains the \(G\) matrix computed from distance a matrix that was computed on a version of \(Y\) where the \(i^{th}\) column has been randomly permuted.

ncores  Integer; if `ncores > 1`, the `parallel` package is used to speed computation. Note: Windows users must set `ncores = 1` because the `parallel` package relies on forking. See `mc.cores` in the `mclapply` function in the `parallel` package for more details.

seed  Integer; sets seed for the permutations of each variable comprising \(Y\) so that results can be replicated.

plot.res  Logical; Indicates whether or not a heat-map of the results should be plotted.

grayscale  Logical; Indicates whether or not the heat-map should be plotted in grayscale.

cex  Multiplier for `cex.axis`, `cex.lab`, `cex.main`, and `cex` that are passed to the plotted result.

y.las  Orientation of labels for the outcome items. Defaults to vertical (2). Value of 1 prints horizontal labels, and is only recommended if the multivariate outcome is comprised of few variables.

Details

See McArtor et al. (2017) for a detailed description of how delta is computed. Note that it is a relative measure of effect, quantifying which effects are strong (high values of delta) and weak (low values of delta) within a single analysis, but estimates of delta cannot be directly compared across different datasets.
There are two options for using this function. The first option is to specify the predictor matrix X, the outcome matrix Y, the distance type dtype (supported by "dist" in R), and number of iterations niter. This option conducts the permutation of each Y-item niter times (to average out random association in each permutation) and reports the median estimates of delta over the niter reps.

The second option is to specify X, G, and G.list, a list of G matrices where the permutation has already been done for each item comprising Y. The names of the elements in G.list should correspond to the names of the variables that were permuted. This option is implemented so that delta can be computed when MDMR is being used in conjunction with distance metrics not supported by dist.

**Value**

A data frame whose rows correspond to the omnibus effects and the effect of each individual predictor (conditional on the rest), and whose columns correspond to each outcome variable whose effect sizes are being quantified. If plot.res = TRUE, a heat-map is plotted of this data frame to easily identify the strongest effects. Note that the heatmap is partitioned into the omnibus effect (first row) and pair-wise effects (remaining rows), because otherwise the omnibus effect would dominate the heatmap.

**Author(s)**

Daniel B. McArtor (dmcartor@gmail.com) [aut, cre]

**References**


**Examples**

```r
data(mdmrdata)
# --- Method 1 --- #
delta(X.mdmr, Y = Y.mdmr, dtype = "euclidean", niter = 1, seed = 12345)

# --- Method 2 --- #
D <- dist(Y.mdmr, method = "euclidean")
G <- gower(D)
q <- ncol(Y.mdmr)
G.list <- vector(mode = "list", length = q)
names(G.list) <- names(Y.mdmr)
for(i in 1:q) {
  Y.shuf <- Y.mdmr
  Y.shuf[,i] <- sample(Y.shuf[,i])
  G.list[[i]] <- gower(dist(Y.shuf, method = "euclidean"))
}
delta(X.mdmr, G = G, G.list = G.list)
```
gower

**Compute Gower’s centered similarity matrix from a distance matrix**

Description

Compute Gower’s centered similarity matrix $G$, which is the matrix decomposed by the MDMR test statistic.

Usage

gower(d.mat)

Arguments

d.mat Symmetric distance matrix (or R distance object) computed from the outcome data to be used in MDMR.

Value

G Gower’s centered dissimilarity matrix computed from D.

Author(s)

Daniel B. McArtor (dmcartor@gmail.com) [aut, cre]

References


mdmr

**Conduct MDMR with analytic p-values**

Description

mdmr (multivariate distance matrix regression) is used to regress a distance matrix onto a set of predictors. It returns the test statistic, pseudo R-square statistic, and analytic p-values for all predictors jointly and for each predictor individually, conditioned on the rest.

Usage

mdmr(X, D = NULL, G = NULL, lambda = NULL, return.lambda = F, start.acc = 1e-20, ncores = 1, perm.p = (nrow(as.matrix(X)) < 200), nperm = 500, seed = NULL)
Arguments

X  A \( n \times p \) matrix or data frame of predictors. Unordered factors will be tested with contrast-codes by default, and ordered factors will be tested with polynomial contrasts. For finer control of how categorical predictors are handled, or if higher-order effects are desired, the output from a call to `model.matrix()` can be supplied to this argument as well.

D  Distance matrix computed on the outcome data. Can be either a matrix or an R `dist` object. Either D or G must be passed to `mdmr()`.

G  Gower’s centered similarity matrix computed from D. Either D or G must be passed to `mdmr()`.

lambda  Optional argument: Eigenvalues of G. Eigendecomposition of large G matrices can be somewhat time consuming, and the theoretical p-values require the eigenvalues of G. If MDMR is to be conducted multiple times on one distance matrix, it is advised to conduct the eigendecomposition once and pass the eigenvalues to `mdmr()` directly each time.

return.lambda  Logical; indicates whether or not the eigenvalues of G should be returned, if calculated. Default is FALSE.

start.acc  Starting accuracy of the Davies (1980) algorithm implemented in the `davies` function in the `CompQuadForm` package (Duchesne & De Micheaux, 2010) that `mdmr()` uses to compute MDMR p-values.

ncores  Integer; if ncores > 1, the `parallel` package is used to speed computation. Note: Windows users must set ncores = 1 because the `parallel` package relies on forking. See `mc.cores` in the `mclapply` function in the `parallel` package for more details.

perm.p  Logical: should permutation-based p-values be computed instead of analytic p-values? Default behavior is TRUE if n < 200 and FALSE otherwise because the analytic p-values depend on asymptotics. for n > 200 and "permutation" otherwise.

nperm  Number of permutations to use if permutation-based p-values are to be computed.

seed  Random seed to use to generate the permutation null distribution. Defaults to a random seed.

Details

This function is the fastest approach to conducting MDMR. It uses the fastest known computational strategy to compute the MDMR test statistic (see Appendix A of McArtor et al., 2017), and it uses fast, analytic p-values.

The slowest part of conducting MDMR is now the necessary eigendecomposition of the G matrix, whose computation time is a function of \( n^3 \). If MDMR is to be conducted multiple times on the same distance matrix, it is recommended to compute eigenvalues of G in advance and pass them to the function rather than computing them every time `mdmr` is called, as is the case if the argument lambda is left NULL.

The distance matrix D can be passed to `mdmr` as either a distance object or a symmetric matrix.
Value

An object with six elements and a summary function. Calling `summary(mdmr.res)` produces a data frame comprised of:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>Value of the corresponding MDMR test statistic</td>
</tr>
<tr>
<td>Numer DF</td>
<td>Numerator degrees of freedom for the corresponding effect</td>
</tr>
<tr>
<td>Pseudo R2</td>
<td>Size of the corresponding effect on the distance matrix</td>
</tr>
<tr>
<td>p-value</td>
<td>The p-value for each effect.</td>
</tr>
</tbody>
</table>

In addition to the information in the three columns comprising `summary(res)`, the `res` object also contains:

| p.prec        | A data.frame reporting the precision of each p-value. If analytic p-values were computed, these are the maximum error bound of the p-values reported by the `davies` function in `CompQuadForm`. If permutation p-values were computed, it is the standard error of each permutation p-value. |
| lambda        | A vector of the eigenvalues of G (if `return.lambda = T`).                   |
| nperm         | Number of permutations used. Will read NA if analytic p-values were computed |

Note that the printed output of `summary(res)` will truncate p-values to the smallest trustworthy values, but the object returned by `summary(res)` will contain the p-values as computed. The reason for this truncation differs for analytic and permutation p-values. For an analytic p-value, if the error bound of the Davies algorithm is larger than the p-value, the only conclusion that can be drawn with certainty is that the p-value is smaller than (or equal to) the error bound. For a permutation test, the estimated p-value will be zero if no permuted test statistics are greater than the observed statistic, but the zero p-value is only a product of the finite number of permutations conducted. The only conclusion that can be drawn is that the p-value is smaller than $1/nperm$.

Author(s)

Daniel B. McArtor (dmcartor@gmail.com) [aut, cre]

References


Examples

# --- The following two approaches yield equivalent results --- #
# Approach 1
data(mdmrdata)
D <- dist(Y.mdmr, method = "euclidean")
res1 <- mdmr(X = X.mdmr, D = D)
summary(res1)

# Approach 2
data(mdmrdata)
D <- dist(Y.mdmr, method = "euclidean")
G <- gower(D)
res2 <- mdmr(X = X.mdmr, G = G)
summary(res2)

mixed.mdmr

Fit Mixed-MDMR models

Description

mixed.mdmr allows users to conduct multivariate distance matrix regression (MDMR) in the context of a (hierarchically) clustered sample without inflating Type-I error rates as a result of the violation of the independence assumption. This is done by invoking a mixed-effects modeling framework, in which clustering/grouping variables are specified as random effects and the covariate effects of interest are fixed effects. The input to mixed.mdmr largely reflects the input of the lmer function from the package lme4 insofar as the specification of random and fixed effects are concerned (see Arguments for details). Note that this function simply controls for the random effects in order to test the fixed effects; it does not facilitate point estimation or inference on the random effects.

Usage

mixed.mdmr(fmla, data, D = NULL, G = NULL, use.ssd = 1,
         start.acc = 1e-20, ncores = 1)

Arguments

fmla A one-sided linear formula object describing both the fixed-effects and random-effects part of the model, beginning with an ~ operator, which is followed by the terms to include in the model, separated by + operators. Random-effects terms are distinguished by vertical bars (|) separating expressions for design matrices from grouping factors. Two vertical bars (||) can be used to specify multiple uncorrelated random effects for the same grouping variable.

data A mandatory data frame containing the variables named in formula.

D Distance matrix computed on the outcome data. Can be either a matrix or an R dist object. Either D or G must be passed to mdmr().
Gower’s centered similarity matrix computed from D. Either D or G must be passed to mixed.mdmr.

use.ssd

The proportion of the total sum of squared distances (SSD) that will be targeted in the modeling process. In the case of non-Euclidean distances, specifying use.ssd to be slightly smaller than 1.00 (e.g., 0.99) can substantially lower the computational burden of mixed.mdmr while maintaining well-controlled Type-I error rates and only sacrificing a trivial amount of power. In the case of Euclidean distances the computational burden of mixed.mdmr is small, so use.ssd should be set to 1.00.

start.acc

Starting accuracy of the Davies (1980) algorithm implemented in the davies function in the CompQuadForm package (Duchesne & De Micheaux, 2010) that mdmr() uses to compute MDMR p-values.

ncores

Integer; if ncores > 1, the parallel package is used to speed computation. Note: Windows users must set ncores = 1 because the parallel package relies on forking. See mc.cores in the mclapply function in the parallel package for more details.

Value

An object with six elements and a summary function. Calling summary(mixed.mdmr.res) produces a data frame comprised of:

Statistic

Value of the corresponding MDMR test statistic

Numerator DF

Numerator degrees of freedom for the corresponding effect

p-value

The p-value for each effect.

In addition to the information in the three columns comprising summary(res), the res object also contains:

p.prec

A data.frame reporting the precision of each p-value. If analytic p-values were computed, these are the maximum error bound of the p-values reported by the davies function in CompQuadForm. If permutation p-values were computed, it is the standard error of each permutation p-value.

Note that the printed output of summary(res) will truncate p-values to the smallest trustworthy values, but the object returned by summary(res) will contain the p-values as computed. The reason for this truncation differs for analytic and permutation p-values. For an analytic p-value, if the error bound of the Davies algorithm is larger than the p-value, the only conclusion that can be drawn with certainty is that the p-value is smaller than (or equal to) the error bound.

Author(s)

Daniel B. McArtor (dmcartor@gmail.com) [aut, cre]

References


Examples

data("clustmdmrdata")

# Get distance matrix
D <- dist(Y.clust)

# Regular MDMR without the grouping variable
mdmr.res <- mdmr(X = X.clust[,1:2], D = D, perm.p = FALSE)

# Results look significant
summary(mdmr.res)

# Account for grouping variable
mixed.res <- mixed.mdmr(~ x1 + x2 + (x1 + x2 | grp),
                        data = X.clust, D = D)

# Significance was due to the grouping variable
summary(mixed.res)

print.mdmr

Print MDMR Object

Description

print method for class mdmr

Usage

## S3 method for class 'mdmr'
print(x, ...)

Arguments

x Output from mdmr
...
Further arguments passed to or from other methods.

Value

p-value Analytic p-values for the omnibus test and each predictor
Author(s)

Daniel B. McArtor (dmcartor@gmail.com) [aut, cre]

print.mixed.mdmr

Print Mixed MDMR Object

Description

print method for class mixed.mdmr

Usage

## S3 method for class 'mixed.mdmr'
print(x, ...)

Arguments

x

Output from mixed.mdmr

...

Further arguments passed to or from other methods.

Value

p-value

Analytic p-values for the omnibus test and each predictor

Author(s)

Daniel B. McArtor (dmcartor@gmail.com) [aut, cre]

summary.mdmr

Summarizing MDMR Results

Description

summary method for class mdmr

Usage

## S3 method for class 'mdmr'
summary(object, ...)

Arguments

object

Output from mdmr

...

Further arguments passed to or from other methods.
Value

Calling \texttt{summary(mdmr.res)} produces a data frame comprised of:

- **Statistic**: Value of the corresponding MDMR test statistic
- **Pseudo R2**: Size of the corresponding effect on the distance matrix
- **p-value**: The p-value for each effect.

In addition to the information in the three columns comprising \texttt{summary(res)}, the \texttt{res} object also contains:

- **p.prec**: A data.frame reporting the precision of each p-value. If analytic p-values were computed, these are the maximum error bound of the p-values reported by the \texttt{davies} function in \texttt{CompQuadForm}. If permutation p-values were computed, it is the standard error of each permutation p-value.
- **lambda**: A vector of the eigenvalues of \(G\) (if \texttt{return.lambda = T}).
- **nperm**: Number of permutations used. Will read \texttt{NA} if analytic p-values were computed.

Note that the printed output of \texttt{summary(res)} will truncate p-values to the smallest trustworthy values, but the object returned by \texttt{summary(res)} will contain the p-values as computed. The reason for this truncation differs for analytic and permutation p-values. For an analytic p-value, if the error bound of the Davies algorithm is larger than the p-value, the only conclusion that can be drawn with certainty is that the p-value is smaller than (or equal to) the error bound. For a permutation test, the estimated p-value will be zero if no permuted test statistics are greater than the observed statistic, but the zero p-value is only a product of the finite number of permutations conducted. The only conclusion that can be drawn is that the p-value is smaller than \(1/nperm\).

Author(s)

Daniel B. McArtor (dmcartor@gmail.com) [aut, cre]

References


Examples

```r
# --- The following two approaches yield equivalent results --- #
# Approach 1
data(mdmrdata)
D <- dist(Y.mdmr, method = "euclidean")
mdmr.res <- mdmr(X = X.mdmr, D = D)
summary(mdmr.res)
```
summarizeixed.mdmr

**Summary Mixed MDMR Results**

**Description**

Summary method for class `mixed.mdmr`

**Usage**

```r
## S3 method for class 'mixed.mdmr'
summary(object, ...)
```

**Arguments**

- `object`: Output from `mixed.mdmr`
- `...`: Further arguments passed to or from other methods.

**Value**

Calling `summary(mdmr.res)` produces a data frame comprised of:

- **Statistic**: Value of the corresponding MDMR test statistic
- **p-value**: The p-value for each effect.

In addition to the information in the three columns comprising `summary(res)`, the `res` object also contains:

- **p.prec**: A data frame reporting the precision of each p-value. If analytic p-values were computed, these are the maximum error bound of the p-values reported by the `davies` function in `CompQuadForm`. If permutation p-values were computed, it is the standard error of each permutation p-value.

Note that the printed output of `summary(res)` will truncate p-values to the smallest trustworthy values, but the object returned by `summary(res)` will contain the p-values as computed. The reason for this truncation differs for analytic and permutation p-values. For an analytic p-value, if the error bound of the Davies algorithm is larger than the p-value, the only conclusion that can be drawn with certainty is that the p-value is smaller than (or equal to) the error bound.

**Author(s)**

Daniel B. McArtor (dmcartor@gmail.com) [aut, cre]
**References**


**Examples**

```r
data("clustmdmrdata")

# Get distance matrix
D <- dist(Y.clust)

# Regular MDMR without the grouping variable
mdmr.res <- mdmr(X = X.clust[,1:2], D = D, perm.p = FALSE)

# Results look significant
summary(mdmr.res)

# Account for grouping variable
mixed.res <- mixed.mdmr(~ x1 + x2 + (x1 + x2 | grp),
                         data = X.clust, D = D)

# Significance was due to the grouping variable
summary(mixed.res)
```

| X.clust | Simulated clustered predictor data to illustrate the Mixed-MDMR function |

**Description**

See `mixed.mdmr`.

**Usage**

X.clust

**Format**

An object of class `data.frame` with 250 rows and 3 columns.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Usage</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>X.mdmr</td>
<td>Simulated predictor data to illustrate the MDMR package.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>See package vignette by calling vignette(&quot;mdmr-vignette&quot;).</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Description</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>See package vignette by calling vignette(&quot;mdmr-vignette&quot;).</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Usage</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>X.mdmr</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Format</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>An object of class matrix with 500 rows and 3 columns.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y.clust</td>
<td>Simulated clustered outcome data to illustrate the Mixed-MDMR function.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>See mixed.mdmr.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Usage</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y.clust</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Format</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>An object of class data.frame with 250 rows and 12 columns.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y.mdmr</td>
<td>Simulated outcome data to illustrate the MDMR package.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>See package vignette by calling vignette(&quot;mdmr-vignette&quot;).</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Usage</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y.mdmr</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Format</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>An object of class matrix with 500 rows and 10 columns.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Index

* datasets
  X.clust, 15
  X.mdmr, 16
  Y.clust, 16
  Y.mdmr, 16

davies, 7, 10
delta, 2, 3
dist, 4, 7, 9
gower, 6
lme4, 9
lmer, 9

mclapply, 4, 7, 10
MDMR (MDMR-package), 2
mdmr, 2, 6
MDMR-package, 2
mixed.mdmr, 2, 9, 12, 14–16

parallel, 4, 7, 10
print.mdmr, 11
print.mixed.mdmr, 12

summary.mdmr, 12
summary.mixed.mdmr, 14

X.clust, 15
X.mdmr, 16

Y.clust, 16
Y.mdmr, 16