

# Package ‘DVDtest’

June 17, 2020

**Type** Package

**Title** Difference between Varying Distributions Test (DVDtest)

**Version** 0.1-1

**Date** 2020-06-16

**Author** Meng Xu, Philip Reiss

**Maintainer** Meng Xu <mxu@campus.haifa.ac.il>

**Description** See DVDtest.

**RoxygenNote** 7.1.0

**License** GPL (>= 2)

**Encoding** UTF-8

**Imports** gamlss, mgcv, parallel, gamlss.dist, ggplot2, reshape2

## R topics documented:

DVDtest-package . . . . .	1
DVDplot . . . . .	2
DVDtest . . . . .	3
<b>Index</b>	<b>7</b>

---

DVDtest-package	<i>Difference between Varying Distributions Test (DVDtest)</i>
-----------------	--

---

## Description

This package contains a series of internal and external functions of Difference between Varying Distributions Test (DVDtest), which tests the pointwise group differences between two varying distributions.

## Author(s)

Meng Xu, Philip Reiss

Maintainer: Meng Xu <mxu@campus.haifa.ac.il>

## See Also

[DVDtest](#)

---

`DVDplot`*Plotting DVDtest*

---

**Description**

Plot a list of the DVDtest-related figures via ggplot2

**Usage**

```
DVDplot(tobj, kxlab = NULL, kylab = NULL, kname = NULL, thhd = 0.05)
```

**Arguments**

<code>tobj</code>	a return test object of <a href="#">DVDtest</a>
<code>kxlab</code>	a title for the x axis, <code>.index</code>
<code>kylab</code>	a title for the y axis, <code>.value</code>
<code>kname</code>	a name for k, e.g. ROI in the references
<code>thhd</code>	a number to mark out the zone of the p-value of interest. Defaults to 0.05

**Details**

Fig `pfig` illustrates a collection of the p value curves among all k. Fig `kfig` illustrates a collection list of the figures with varying distributions among all k, highlighted the zones of small p values in dark, where `ydata1` and `ydata2` mark as red and blue, respectively. The dashed and dotted lines denote the smooth mean function and  $\pm 2$  sigma, respectively.

**Value**

a list of ggplot objects on p value curves and varying distributions

**Note**

Please contact the maintainer if need more details.

**Author(s)**

Meng Xu

**References**

reiss-EMR18.pdf

**See Also**

Examples in [DVDtest](#)

DVDtest

*Difference between Varying Distributions Test (DVDtest)***Description**

Testing the difference of two varying distributions.

**Usage**

```
DVDtest(
  ydata1,
  ydata2,
  nperm = 60,
  grid,
  dist.method = "wass",
  mgcv.gam = TRUE,
  ...,
  exclude = NULL,
  permadj = FALSE,
  mc.cores = 1,
  chunksize = 2,
  seeds = NULL,
  report.every = 10,
  stepdown = FALSE
)
```

**Arguments**

ydata1	a data.frame or a list of data.frame, containing at least 3 columns called .obs, .index and .value which specify which curve the point belongs to (.obs) at which (.index) it was observed and the observed value (.value). See details in the package refund. Other columns are available as well for modeling the varying distributions.
ydata2	same as ydata1.
nperm	a scalar, number of permutation
grid	a vector, evaluation grids of .index
dist.method	the distance measure to be used. This must be one of Wasserstein distance ('wass'), 'L2' distance, 'L1' distance and 'Hellinger'. It defaults to 'wass'.
mgcv.gam	a logical variable, whether to apply <a href="#">gam</a> for estimating distributions, whose parameters are a smooth function of a continuous variable. If FALSE, <a href="#">gamlss</a> is adopted, which could cover a wider range of varying distributions.
...	passed to arguments of <a href="#">gam</a> or <a href="#">gamlss</a> . If mgcv.gam = TRUE, ... should include formula, family and other optional arguments in mgcv::gam. Otherwise, ... passed to arguments inside of <a href="#">gamlss</a> . See Examples for details.
exclude	passed to exclude inside of predict.gam in case mgcv.gam = TRUE.
permadj	a logical variable, whether to adjust the permuted data to cover the entire range, esp. in case of sparsity. Defaults to FALSE.
mc.cores	passed to mc.cores inside of <a href="#">mclapply</a> (not available on Windows unless mc.cores = 1).

chunksize	a scalar, number of permutations per core/thread when using <code>mcapply</code> . It defaults to 2.
seeds	a scalar, to set the seed for the permutation via <code>set.seed(seeds)</code> . It defaults to <code>NULL</code> .
report.every	a scalar, to report the current number of permutations.
stepdown	a logical variable, which denotes to apply the step-down min P.

### Value

<code>.index</code>	a vector, evaluation grids.
<code>pval</code>	a vector or matrix of (adjusted) p values.
<code>vdparam</code>	a list of paramters of varying distributions.

### Note

- If `ydata1` and `ydata2` are lists of data.frames, the lengths of two lists must be the same.
- If `mgcv.gam` is `TRUE`, ... and `exclue` are `NULL` (default settings), then they both default to `formula <- list(.value ~ s(.index) + s(.obs, bs = "re"), ~ s(.index))`, `family = gauss()` and `exclude <- "s(.obs)"`, repectively.
- Normal distribution (`gauss()`) in `mgcv::gam` is supported. And `gamlss.family` is supported as well by `DVDtest` for fitting a GAMLSS-type varying distributions with various types of random effect. Note that the permuted data may not match some specific distributions during the permutation.

### Author(s)

Meng Xu, Philip Reiss

### References

reiss-EMR18.pdf

Wood, S. N., Pya, N., & Saffken, B. (2016). Smoothing parameter and model selection for general smooth models. *Journal of the American Statistical Association*, 111(516), 1548-1563.

Rigby, R. A., & Stasinopoulos, D. M. (2005). Generalized additive models for location, scale and shape. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 54(3), 507-554.

### Examples

```
library(DVDtest)
## Data Generation ##
p <- 6
mu1 <- function(t) 0.2*(p-1)*sin(pi*t)+t+1
mu2 <- function(t) -0.2*(p-1)*sin(pi*t)+t+1
sig1 <- function(t) t+1
sig2 <- sig1
nperson <- 10
fun1 <- function(t) rnorm(nperson, mu1(t), sig1(t))
fun2 <- function(t) rnorm(nperson, mu2(t), sig2(t))
tp <- seq(0,1,,10)
data1 <- sapply(tp,fun1)
data2 <- sapply(tp,fun2)
```

```

library(reshape2)
colnames(data2) <- colnames(data1) <- tp
rownames(data2) <- 1:nperson+2*nperson
dg1 <- melt(data1)
dg2 <- melt(data2)
colnames(dg1) <- colnames(dg2) <- c('.obs', '.index', '.value')
# library(ggplot2)
# ggplot() + geom_line(data = dg1, aes(x = .index, y = .value, group = .obs))
# + geom_line(data = dg2, aes(x = .index, y = .value, group = .obs))

ngrid <- 50
ev.grid <- seq(0, 1, , ngrid)
nperm. <- 40

####Estimated with mgcv::gam
library(mgcv)
simu.test1 <- DVDtest(dg1, dg2, nperm., ev.grid)

####Estimated with gamlss::gamlss
library(gamlss)
simu.test2 <- DVDtest(dg1, dg2, nperm., ev.grid, formula = .value ~ pb(.index),
                      sigma.formula = ~pb(.index), random = ~1|.obs, family = NO, mgcv.gam=FALSE)

####Plot
simu.figs <- DVDplot(simu.test1)
simu.figs$pfig
simu.figs$kf1fig[[1]]

####Non-normal case
## Not run:
p <- 6
mu1 <- function(t) 0.2*(p-1)*sin(pi*t)+t+1
mu2 <- function(t) -0.2*(p-1)*sin(pi*t)+t+1
sig1 <- function(t) t+1
sig2 <- sig1
nu1 <- function(t) t+1
nu2 <- nu1
nperson <- 10

fun1 <- function(t) rGG(nperson, mu1(t), sig1(t), nu1(t))
fun2 <- function(t) rGG(nperson, mu2(t), sig2(t), nu2(t))
tp <- seq(0, 1, , 10)
data1 <- sapply(tp, fun1)
data2 <- sapply(tp, fun2)

library(reshape2)
colnames(data2) <- colnames(data1) <- tp
rownames(data2) <- 1:nperson+nperson
dg1 <- melt(data1)
dg2 <- melt(data2)
colnames(dg1) <- colnames(dg2) <- c('.obs', '.index', '.value')

ngrid <- 50
ev.grid <- seq(0, 1, , ngrid)
nperm. <- 40

```

```
simu.test3 <- DVDtest(dg1, dg2, nperm.,ev.grid, formula = .value ~ pb(.index),  
                      sigma.formula = ~pb(.index),  
                      nu.formula= ~pb(.index), seeds=123,  
                      random = ~1|.obs, family = GG, mgcv.gam = FALSE)  
  
## End(Not run)
```

# Index

- \* **ggplot**
  - DVDplot, [2](#)
- \* **permutation,**
  - DVDtest, [3](#)
- \* **pointwise**
  - DVDtest, [3](#)

DVDplot, [2](#)  
DVDtest, [1](#), [2](#), [3](#)  
DVDtest-package, [1](#)

gam, [3](#)  
gamlss, [3](#)  
gamlss.family, [4](#)

mclapply, [3](#)