

Package ‘hogsvdR’

October 21, 2017

Version 0.1

Date 2017-10-17

Title Higher-Order Generalized Singular Value Decomposition

Author Nikolas Barkas [aut, cre]

Depends R (>= 3.1.0)

Imports MASS, Rcpp

Description Implementation of higher order generalized singular value decomposition (HO GSVD). Based on Ponnappalli, Saunders, etal (2011) <doi:10.1371/journal.pone.0028072>.

License GPL (>= 2)

URL <https://github.com/barkasn/hogsvdR>

BugReports <https://github.com/barkasn/hogsvdR/issues>

RoxygenNote 6.0.1

LinkingTo Rcpp, RcppArmadillo

Suggests testthat

NeedsCompilation yes

Maintainer Nikolas Barkas <Nikolaos_Barkas@hms.harvard.edu>

Repository CRAN

Date/Publication 2017-10-21 14:48:24 UTC

R topics documented:

calcNormS	2
calcNormS.R	2
hogsvd	2
hogsvd.rArmadillo	4
hogsvd.rsimple	4

Index	5
--------------	----------

calcNormS	<i>Calculate normalised S matrix</i>
-----------	--------------------------------------

Description

Calculate normalised S matrix

Usage

```
calcNormS(D, ncols)
```

Arguments

D	list of data matrices
ncols	number of columns of input matrices

calcNormS.R	<i>Calculate the normalised S matrix in R, for internal use</i>
-------------	---

Description

Calculate the normalised S matrix in R, for internal use

Usage

```
calcNormS.R(D)
```

Arguments

D	a list of matrices
---	--------------------

hogsvd	<i>Compute the Higher-order generalised singular value decomposition (HOGSVD) of a list of matrices</i>
--------	---

Description

Compute the Higher-order generalised singular value decomposition (HOGSVD) of a list of matrices

Usage

```
hogsvd(D, method = "arma")
```

Arguments

`D` a list of matrices to compute the GSVD decomposition on

`method` specification of internal function to use to compute HOGSVD, 'arma' or 'rsimple'

Value

A list of U, Sigma and V. U and Sigma are lists of matrices

Examples

```
# Generate 3 matrices to run example on
N <- 3
nrow <- c(10,10,10)
ncol <- 10
s <- 1:N
D <- lapply(s, function(x) {matrix(rnorm(n=nrow[x]*ncol,mean = 0, sd =10),nrow[x],ncol)})

# Perform HO GSVD on the example
res <- hogsvd(D)

# Inspect result
str(res)

# The first U matrix corresponding to D[[1]]
res$U[[1]]

# The first S diagonal matrix corresponding to D[[1]]
res$S[[1]]

# The shared V matrix
res$V

# Reconstruct the original matrices
D.reconstruct <- lapply(1:N, function(n) {
  res$U[[n]] %*% diag(res$Sigma[[n]]) %*% t(res$V)
})

# Now repeat with the slow algorithm
res.slow <- hogsvd(D, method = 'rsimple')
D.reconstruct.slow <- lapply(1:N, function(n) {
  res.slow$U[[n]] %*% diag(res.slow$Sigma[[n]]) %*% t(res.slow$V)
})
```

hogsvd.rArmadillo	<i>Compute the Higher-order generalised singular value decomposition (HOGSVD) of a list of matrices with RcppArmadillo</i>
-------------------	--

Description

Compute the Higher-order generalised singular value decomposition (HOGSVD) of a list of matrices with RcppArmadillo

Usage

```
hogsvd.rArmadillo(D)
```

Arguments

D a list of matrices to compute the GSVD decomposition on

Value

A list of U, Sigma, V, Lambda and S. U and Sigma are lists of matrices

hogsvd.rsimple	<i>Compute the Higher-order generalised singular value decomposition (HOGSVD) of a list of matrices without Armadillo</i>
----------------	---

Description

Compute the Higher-order generalised singular value decomposition (HOGSVD) of a list of matrices without Armadillo

Usage

```
hogsvd.rsimple(D)
```

Arguments

D a list of matrices to compute the GSVD decomposition on

Value

A list of U, Sigma, V, Lambda and S. U and Sigma are lists of matrices

Index

`calcNormS`, [2](#)
`calcNormS.R`, [2](#)

`hogsvd`, [2](#)
`hogsvd.rArmadillo`, [4](#)
`hogsvd.rsimple`, [4](#)