Package 'iTensor'

October 13, 2022

Type Package

Title ICA-Based Matrix/Tensor Decomposition

Version 0.99.0

Description Some functions for performing ICA, MICA, and Multilinear ICA are implemented. ICA, MICA, and Multilinear ICA extract statistically independent components from single matrix, multiple matrices, and single tensor, respectively.
For the details of these methods, see the reference section of GitHub README.md <https://github.com/rikenbit/iTensor>.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

URL https://github.com/rikenbit/mwTensor

RoxygenNote 7.1.2

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Imports MASS, methods, rTensor, stats

NeedsCompilation no

Author Koki Tsuyuzaki [aut, cre]

Maintainer Koki Tsuyuzaki <k.t.the-answer@hotmail.co.jp>

Repository CRAN

Date/Publication 2022-06-14 10:10:02 UTC

R topics documented:

ICA *Independent Component Analysis The input data is assumed to be a matrix. ICA decomposes the matrix and extract the components that are statistically independent each other.*

Description

Independent Component Analysis The input data is assumed to be a matrix. ICA decomposes the matrix and extract the components that are statistically independent each other.

Usage

```
ICA(
  X,
  J,
  algorithm = c("FastICA", "InfoMax", "ExtInfoMax"),
  num.iter = 100,
  thr = 1e-10,
  nonlinear_func = c("tanh", "exp", "kurtosis"),
  learning_rate = 1,
  verbose = FALSE
)
```

Arguments

X	A matrix
J	Rank parameter to decompose
algorithm	The decomposition algorithm (Default: "FastICA")
num.iter	The number of iteration
thr	The threshold to terminate the iteration (Default: 1E-10)
nonlinear_func	The function used in FastICA (Default: "tanh")
learning_rate	The learning rate used in InfoMax or ExtInfoMax
verbose	Verbose option

Value

A list containing the result of the decomposition

Examples

```
X <- matrix(runif(100*200), nrow=100, ncol=200)
J <- 5
out.FastICA <- ICA(X, J=J, algorithm="FastICA")
out.InfoMax <- ICA(X, J=J, algorithm="InfoMax")
out.ExtInfoMax <- ICA(X, J=J, algorithm="ExtInfoMax")
```

MICA — *Multimodal independent component analysis The input datasets are assumed to be two matrices sharing the column space. MICA decomposes the matrices simutanously and extracts the components that maximizes the mutual information between the components.*

Description

Multimodal independent component analysis The input datasets are assumed to be two matrices sharing the column space. MICA decomposes the matrices simutanously and extracts the components that maximizes the mutual information between the components.

Usage

```
MICA(
  X,
  Y,
  J,
  eta = 1000 * 1e-04,
  verbose = FALSE,
  mu = 50 * 1e-04,
  gamma_ts = 1
)
```

Arguments

X	A matrix sharing the column space with Y (??? x N)
Y	A matrix sharing the column space with X (??? x N)
J	The rank parameter to decompose the matrices
eta	A learning rate parameter of stochastic gradient descent
verbose	Verbose option
mu	A learning rate parameter of stochastic gradient descent
gamma_ts	Weighting factor for dependence on independence

Value

A list containing the result of the decomposition

Examples

```
X <- array(runif(10*20), dim=c(10,20))
Y <- array(runif(15*20), dim=c(15,20))
J <- 20
out <- MICA(X, Y, J=J)
```

`MultilinearICA` *Multilinear independent component analysis The input object is assumed to be a Tensor object defined by rTensor package. In MultilinearICA, ICA function is performed in each mode of the tensor.*

Description

Multilinear independent component analysis The input object is assumed to be a Tensor object defined by rTensor package. In MultilinearICA, ICA function is performed in each mode of the tensor.

Usage

```
MultilinearICA(
  X,
  Js = c(3, 3, 3),
  modes = 1:3,
  algorithm = c("FastICA", "InfoMax", "ExtInfoMax")
)
```

Arguments

`X`	An rTensor object
`Js`	A vector to specify the rank in each mode (Default: c(3,3,3))
`modes`	A vector to specify which modes are decomposed (Default: 1:3)
`algorithm`	The algorithm to decompose the input tensor in each mode (Default: "FastICA")

Value

A list containing the result of the decomposition

Examples

```
library("rTensor")
arrX <- array(runif(10*20*30), dim=c(10,20,30))
X <- as.tensor(arrX)
Js <- c(2,3,4)
out <- MultilinearICA(X, Js=Js)
```

Index