

Package ‘icamix’

April 17, 2017

Type Package

Title Estimation of ICA Mixture Models

Version 1.0.6

Date 2017-04-16

Author Xiaotian Zhu, David R. Hunter

Maintainer Xiaotian Zhu <xzhu@natera.com>

Description Provides R functions which facilitate the estimation of ICA mixture models. We have developed and implemented the NSMM-ICA algorithm that currently integrates npEM and Fast-ICA for non-parametric estimation of ICA mixture models (Zhu, X., & Hunter, D.R., 2016 <DOI:10.1080/10485252.2016.1225049> and Zhu, X., & Hunter, D.R., 2015 <arXiv:1511.04801> supervised clustering.

Suggests mixtools

License GPL-2 | GPL-3

Imports Rcpp (>= 0.11.5)

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-04-17 16:26:46 UTC

R topics documented:

icamix-package	2
ATRANSDENSITY	2
CLASSDIFFRATE	3
EMFASTICAALG	4
ESTIMATEDMEMBER	5
plot.EMFASTICAALG	6
plot.summary.EMFASTICAALG	7
print.EMFASTICAALG	7
print.summary.EMFASTICAALG	8
summary.EMFASTICAALG	9

Index	10
--------------	-----------

icamix-package

icamix: Estimation for ICA Mixture Model

Description

Provides R functions which facilitate the estimation of ICA mixture models. This is a new tool for unsupervised clustering.

Details

Package: icamix
Type: Package
Version: 1.0.6
Date: 2017-04-16
License: GPL-2 | GPL-3

We have developed the NSMM-ICA algorithm which is implemented in the main function EMFASTICAALG that currently integrates npEM and Fast-ICA for non-parametric estimation of ICA mixture models.

Author(s)

Xiaotian Zhu, David R. Hunter
Maintainer: Xiaotian Zhu <xzhu@natera.com>

References

Zhu, Xiaotian, and David R. Hunter. "Theoretical grounding for estimation in conditional independence multivariate finite mixture models." *Journal of Nonparametric Statistics* 28.4 (2016): 683-701.

Zhu, Xiaotian, and David R. Hunter. "Clustering Via Finite Nonparametric ICA Mixture Models." *arXiv preprint arXiv:1510.08178*

ATRANSDENSITY*ATRANSDENSITY*

Description

A function evaluates density of linearly transformed random vector on a given grid. It is used in processing EMFASTICAALG object to obtain density estimation of the mixture components.

Usage

```
ATRANSDENSITY(grid, A, f)
```

Arguments

grid	A matrix whose columns store the grid points.
A	Matrix for the linear transformation.
f	Density function before the linear transformation.

Value

answer	Matrix of the same size as grid, with each element being the evaluated linear transformed density at the corresponding grid point.
--------	--

Examples

```
## An example that evaluates the 2-D uniform distribution on a linear transformation of [1,3]x[1,3]
## flind is the density of the uniform distribution on [1,3]^r
flind <- function(grid){# mixture component 1 original signal density function
  n <- ncol(grid)
  r <- nrow(grid)
  answer <- rep(1,n)
  for(i in 1:n){
    for(j in 1:r){
      answer[i] <- answer[i] * (grid[j,i] >= 1 & grid[j,i] <= 3) / 2
    }
  }
  answer
}

A <- matrix(c(6, 9, -12, 15), 2, 2, byrow = FALSE)

gridpoints <- t(as.matrix(expand.grid(seq(-32,12,2),seq(18,80,2))))

f1trans <- ATRANSDENSITY(gridpoints, A, flind)

plot(t(gridpoints),col=(f1trans>0))
```

 CLASSDIFFRATE

 CLASSDIFFRATE

Description

A function calculates classification difference rate between two factors. It is used in interpreting info stored in EMFASTICA object.

Usage

```
CLASSDIFFRATE(factor1, factor2)
```

Arguments

factor1	First factor.
factor2	Second factor of the same length as the First factor.

Value

answer	The percentage of instances when factor1[i] is not equal factor2[i].
--------	--

Examples

```
## An example evaluates the classification difference rate
## between two classification results in the form of factors
fac1<-factor(c(1,4,2,3,1,1,3,3,1,2,2,1))
fac2<-factor(c(3,1,2,2,1,2,4,3,2,3,1,1))
CLASSDIFFRATE(fac1, fac2)
```

EMFASTICAALG

EMFASTICAALG

Description

An R wrapper for carrying out NSMM-ICA on nonparametric multivariate ICA mixture data.

Usage

```
EMFASTICAALG(DataMatrix, numCluster, h = 0, maxiter = 300,
  icaiter = 150, tol = 1e-06, verb = TRUE, combine = TRUE,
  seednum = 82196, ...)
```

Arguments

DataMatrix	A matrix of which the rows are data entries. Its dimension is n by r.
numCluster	Predetermined number of mixing components m.
h	Bandwidth. If h is set equal zero (default), iterative bandwidth selection will be used.
maxiter	Maximum number of iterations. Default is 300.
icaiter	Maximum number of ICA iterations in each step. Default is 150.
tol	Threshold that defines convergence (of the outer loop). Default is 1e-6.
verb	TRUE (default) or FALSE indicating whether to print out info at each iteration.
combine	TRUE (default) or FALSE indicating whether to implement the ICA step.
seednum	Seed number (default is 82196) used in kmeans before 1st iteration.
...	

Value

The returned value is an EMFASTICAALG object which consists of a list of items:

<code>\$InputData</code>	A matrix of which the columns are data entries. Its dimension is r by n .
<code>\$Lambdas</code>	A matrix where rows store estimated mixing weights from each iteration.
<code>\$Wmtrs</code>	List of r by r unmixing matrices for each of the m clusters.
<code>\$WUnmixZ</code>	List of unmixing matrices for whitened data for each of the m clusters.
<code>\$OriginalSignals</code>	List of Recovered ICA components for each of the m clusters.
<code>\$ProductDensity</code>	m by n matrix where each row stores the estimated density value of the observed data points for each of the m clusters.
<code>\$MembershipProbs</code>	n by m matrix where each row stores the component membership probabilities of the corresponding data point.
<code>\$ObjValue</code>	Vector holding values of data loglikelihood.
<code>\$ICABandWidth</code>	Matrix holding choices of bandwidth for original signals.
<code>\$call</code>	The function call that results in the returned object.
<code>\$time</code>	Computing time elapsed in second.

Examples

```
## An Example that runs the NSMM-ICA algorithm on Cohen's tone data
data(tonedata, package="mixtools")
b <- EMFASTICAALG(tonedata, 2, h=0, tol=1e-8)
```

 ESTIMATEDMEMBER

ESTIMATEDMEMBER

Description

A function calculates estimated class membership from an EMFASTICAALG object.

Usage

```
ESTIMATEDMEMBER(rstICAMIX)
```

Arguments

`rstICAMIX` An EMFASTICAALG object.

Value

`estimatedmember`

A factor with levels representing the estimated classes.

Examples

```
## An Example that runs the NSMM-ICA algorithm on Cohen's tone data
data(tonedata, package="mixtools")

b <- EMFASTICAALG(tonedata, 2, h=0, tol=1e-8)
estimatedgroup <- ESTIMATEDMEMBER(b) # estimated species info
```

```
plot.EMFASTICAALG      plot.EMFASTICAALG
```

Description

plot method for class EMFASTICAALG.

Usage

```
## S3 method for class 'EMFASTICAALG'
plot(x, vec1 = c(1:2), vec2 = c(1:2), ...)
```

Arguments

x	An EMFASTICAALG object.
vec1	An integer vector of length two specifying the coordinates with respect to which the data is scatter plotted.
vec2	An integer vector of length two specifying the coordinates with respect to which the original signal for each mixture component is scatter plotted.
...	

Value

Returned (invisibly) is the full value of x itself.

Examples

```
## An Example that runs the NSMM-ICA algorithm on Cohen's tone data
data(tonedata, package="mixtools")

b <- EMFASTICAALG(tonedata, 2, h=0, tol=1e-8)
plot(b)
```

```
plot.summary.EMFASTICAALG  
      plot.summary.EMFASTICAALG
```

Description

plot method for class summary.EMFASTICAALG.

Usage

```
## S3 method for class 'summary.EMFASTICAALG'  
plot(x, vec1 = c(1:2), vec2 = c(1:2), ...)
```

Arguments

x	An summary.EMFASTICAALG object.
vec1	An integer vector of length two specifying the coordinates with respect to which the data is scatter plotted.
vec2	An integer vector of length two specifying the coordinates with respect to which the original signal for each mixture component is scatter plotted.
...	

Value

Returned (invisibly) is the full value of x itself.

Examples

```
## An Example that runs the NSMM-ICA algorithm on Cohen's tone data  
data(tonedata, package="mixtools")  
  
b <- EMFASTICAALG(tonedata, 2, h=0, tol=1e-8)  
plot(summary(b))
```

```
print.EMFASTICAALG      print.EMFASTICAALG
```

Description

`print` method for class EMFASTICAALG.

Usage

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

Arguments

x An EMFASTICA object.
...

Value

Returned (invisibly) is the full value of x itself.

Examples

```
## An Example that runs the NSMM-ICA algorithm on Cohen's tone data
data(tonedata, package="mixtools")

b <- EMFASTICAALG(tonedata, 2, h=0, tol=1e-8)
print(b)
```

```
print.summary.EMFASTICAALG
```

```
print.summary.EMFASTICAALG
```

Description

[print](#) method for class `summary.EMFASTICAALG`.

Usage

```
## S3 method for class 'summary.EMFASTICAALG'
print(x, ...)
```

Arguments

x An `summary.EMFASTICA` object.
...

Value

Returned (invisibly) is the full value of x itself.

Examples

```
## An Example that runs the NSMM-ICA algorithm on Cohen's tone data
data(tonedata, package="mixtools")

b <- EMFASTICAALG(tonedata, 2, h=0, tol=1e-8)
print(summary(b))
```

summary.EMFASTICAALG *summary.EMFASTICAALG*

Description

[summary](#) method for class EMFASTICAALG.

Usage

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

Arguments

object An EMFASTICAALG object.
 ...

Value

The returned value is a "summary.EMFASTICAALG" object which consists a list:

<code>\$inputData</code>	A matrix of which the columns are data entries. Its dimension is r by n .
<code>\$originSig</code>	List of Recovered ICA components for each of the m clusters.
<code>\$call</code>	The function call which results in the corresponding EMFASTICAALG object.
<code>\$time</code>	Computing time elaped in second.
<code>\$numIter</code>	Total number of iterations.
<code>\$lastObj</code>	Objective function value from the last iteration.
<code>\$compMeans</code>	Means of each mixture component.
<code>\$compVars</code>	Covariances of each mixture component.
<code>\$numObs</code>	Total number of observations.
<code>\$numAtr</code>	Dimension of data points.
<code>\$numCls</code>	Number of mixture components.
<code>\$estWts</code>	Estimated mixing weights.
<code>\$estCls</code>	A factor whose levels represent estimated class membership.
<code>\$dataLkLhd</code>	A vector storing data loglikelihood from each iteration.

Examples

```
## An Example that runs the NSMM-ICA algorithm on Cohen's tone data
data(tonedata, package="mixtools")

b <- EMFASTICAALG(tonedata, 2, h=0, tol=1e-8)
summary(b)
```

Index

*Topic **\textasciitildekwd1**

ATRANSDENSITY, [2](#)
CLASSDIFFRATE, [3](#)
EMFASTICAALG, [4](#)
ESTIMATEDMEMBER, [5](#)
plot.EMFASTICAALG, [6](#)
plot.summary.EMFASTICAALG, [7](#)
print.EMFASTICAALG, [7](#)
print.summary.EMFASTICAALG, [8](#)

*Topic **\textasciitildekwd2**

ATRANSDENSITY, [2](#)
CLASSDIFFRATE, [3](#)
EMFASTICAALG, [4](#)
ESTIMATEDMEMBER, [5](#)
plot.EMFASTICAALG, [6](#)
plot.summary.EMFASTICAALG, [7](#)
print.EMFASTICAALG, [7](#)
print.summary.EMFASTICAALG, [8](#)

*Topic **package**

icamix-package, [2](#)

ATRANSDENSITY, [2](#)

CLASSDIFFRATE, [3](#)

EMFASTICAALG, [4](#)

ESTIMATEDMEMBER, [5](#)

icamix (icamix-package), [2](#)

icamix-package, [2](#)

plot.EMFASTICAALG, [6](#)

plot.summary.EMFASTICAALG, [7](#)

print, [7](#), [8](#)

print.EMFASTICAALG, [7](#)

print.summary.EMFASTICAALG, [8](#)

summary, [9](#)

summary.EMFASTICAALG, [9](#)