Package ‘ClusVis’

June 24, 2019

Type Package
Title Gaussian-Based Visualization of Gaussian and Non-Gaussian Model-Based Clustering
Version 1.2.0
Author Christophe Biernacki [aut],
Matthieu Marbac [aut, cre],
Vincent Vandewalle [aut]
Maintainer Matthieu Marbac <matthieu.marbac@gmail.com>
Description Gaussian-Based Visualization of Gaussian and Non-Gaussian Model-Based Clustering done on any type of data. Visualization is based on the probabilities of classification.
License GPL (>= 2)
Imports Rcpp, MASS, parallel, mgcv, mvtorm, Rmixmod, VarSelLCM (>= 2.1)
LinkingTo Rcpp, RcppArmadillo, mvtorm
ByteCompile true
Encoding UTF-8
LazyLoad yes
LazyData true
Collate 'clusvis.R' 'estimation.R' 'smartinit.R' 'RcppExports.R'
'plot.R' 'clusvisimod.R' 'clusvisvarellcm.R' 'modessearch.R'
Depends R (>= 3.4)
RoxygenNote 6.1.1
NeedsCompilation yes
Repository CRAN
Date/Publication 2019-06-24 08:30:15 UTC

R topics documented:

ClusVis-package .................................................. 2
clusvis ............................................................. 4
Gaussian-Based Visualization of Gaussian and Non-Gaussian Model-Based Clustering.

Description

The main function for parameter inference is `clusvis`. Moreover, specific functions `clusvisVarSelLCM` and `clusvisMixmod` are implemented to visualize the results of the R package VarSelLCM and Rmixmod. After parameter inference, visualization is done with function `plotDensityClusVisu`.

Details

<table>
<thead>
<tr>
<th>Package</th>
<th>ClusVis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Package</td>
</tr>
<tr>
<td>Version</td>
<td>1.1.0</td>
</tr>
<tr>
<td>Date</td>
<td>2018-04-18</td>
</tr>
<tr>
<td>License</td>
<td>GPL-3</td>
</tr>
<tr>
<td>LazyLoad</td>
<td>yes</td>
</tr>
</tbody>
</table>

Author(s)

Biernacki, C. and Marbac, M. and Vandewalle, V.

Examples

```r
## Not run:

## First example: R package Rmixmod
# Package loading
require(Rmixmod)

data("congress")
# Model-based clustering with 4 components
set.seed(123)
res <- mixmodCluster(congress[-1], 4, strategy = mixmodStrategy(nbTryInInit = 500, nbTry=25))

# Inference of the parameters used for results visualization
```
# (specific for Rmixmod results)
# It is better because probabilities of classification are generated
# by using the model parameters
resvisu <- clusvisMixmod(res)

# Component interpretation graph
plotDensityClusVisu(resvisu)

# Scatter-plot of the observation memberships
plotDensityClusVisu(resvisu, add.obs = TRUE)

## Second example: R package Rmixmod
# Package loading
require(Rmixmod)

# Data loading (categorical data)
data(birds)

# Model-based clustering with 3 components
resmixmod <- mixmodCluster(birds, 3)

# Inference of the parameters used for results visualization (general approach)
# Probabilities of classification are not sampled from the model parameter,
# but observed probabilities of classification are used for parameter estimation
resvisu <- clusvis(log(resmixmod@bestResult@proba),
                   resmixmod@bestResult@parameters@proportions)

# Inference of the parameters used for results visualization
# (specific for Rmixmod results)
# It is better because probabilities of classification are generated
# by using the model parameters
resvisu <- clusvisMixmod(resmixmod)

# Component interpretation graph
plotDensityClusVisu(resvisu)

# Scatter-plot of the observation memberships
plotDensityClusVisu(resvisu, add.obs = TRUE)

## Third example: R package VarSelLCM
# Package loading
require(VarSelLCM)

# Data loading (categorical data)
data("heart")
# Model-based clustering with 3 components
res <- VarSelCluster(heart[, -13], 3)

# Inference of the parameters used for results visualization
# (specific for VarSelLCM results)
# It is better because probabilities of classification are generated
# by using the model parameters
clusvis <- clusvisVarSelLCM(resvisu)

# Component interpretation graph
plotDensityClusVisu(resvisu)

# Scatter-plot of the observation memberships
plotDensityClusVisu(resvisu, add.obs = TRUE)

## End(Not run)

---

clusvis  

*This function estimates the parameters used for visualization*

**Description**

This function estimates the parameters used for visualization

**Usage**

clusvis(logtik.estim, prop = rep(1/ncol(logtik.estim), ncol(logtik.estim)), logtik.obs = NULL, maxit = 10^3, nbrandomInit = 12, nbcpu = 1)

**Arguments**

- **logtik.estim** matrix. It contains the probabilities of classification used for parameter inference (should be sampled from the model parameter or computed from the observations).
- **prop** vector. It contains the class proportions (by default, classes have same proportion).
- **logtik.obs** matrix. It contains the probabilities of classification of the clustered sample. If missing, logtik.estim is used.
- **maxit** numeric. It limits the number of iterations for the Quasi-Newton algorithm (default 1000).
- **nbrandomInit** numeric. It defines the number of random initialization of the Quasi-Newton algorithm.
- **nbcpu** numeric. It specifies the number of CPU (only for linux)

**Value**

Returns a list
Examples

```r
## Not run:

## First example: R package Rmixmod
# Package loading
require(Rmixmod)

# Data loading (categorical data)
data("congress")
# Model-based clustering with 4 components
set.seed(123)
res <- mixmodCluster(congress[, -1], 4, strategy = mixmodStrategy(nbTryInInit = 500, nbTry = 25))

# Inference of the parameters used for results visualization
# (specific for Rmixmod results)
# It is better because probabilities of classification are generated
# by using the model parameters
resvisu <- clusvisMixmod(res)

# Component interpretation graph
plotDensityClusVisu(resvisu)

# Scatter-plot of the observation memberships
plotDensityClusVisu(resvisu, add.obs = TRUE)

## Second example: R package Rmixmod
# Package loading
require(Rmixmod)

# Data loading (categorical data)
data(birds)

# Model-based clustering with 3 components
resmixmod <- mixmodCluster(birds, 3)

# Inference of the parameters used for results visualization (general approach)
# Probabilities of classification are not sampled from the model parameter,
# but observed probabilities of classification are used for parameter estimation
resvisu <- clusvis(log(resmixmod@bestResult@proba),
                   resmixmod@bestResult@parameters@proportions)

# Inference of the parameters used for results visualization
# (specific for Rmixmod results)
# It is better because probabilities of classification are generated
# by using the model parameters
resvisu <- clusvisMixmod(resmixmod)

# Component interpretation graph
plotDensityClusVisu(resvisu)

# Scatter-plot of the observation memberships
```
This function estimates the parameters used for visualization of model-based clustering performed by R package Rmixmod. To achieve the parameter inference, it automatically samples probabilities of classification from the model parameters.

**Usage**

```r
clusvisMixmod(mixmodResult, sample.size = 5000, maxit = 10^3, nrandomInit = 4 * mixmodResult@bestResult@nbCluster, nbcpu = 1, loccont = NULL)
```

**Arguments**

- `mixmodResult` [MixmodCluster] It is an instance of class MixmodCluster returned by function mixmodCluster of R package Rmixmod.
- `sample.size` numeric. Number of probabilities of classification sampled for parameter inference.
clusvisMixmod

maxit numeric. It limits the number of iterations for the Quasi-Newton algorithm (default 1000).

nbrandomInit numeric. It defines the number of random initialization of the Quasi-Newton algorithm.

nbcpu numeric. It specifies the number of CPU (only for linux).

loccont numeric. Index of the column containing continuous variables (only for mixed-type data).

Value

Returns a list

Examples

## Not run:

```r
## First example: R package Rmixmod
# Package loading
require(Rmixmod)

# Data loading (categorical data)
data("congress")
# Model-based clustering with 4 components
set.seed(123)
res <- mixmodCluster(congress[-1], 4, strategy = mixmodStrategy(nbTryInInit = 500, nbTry=25))

# Inference of the parameters used for results visualization
# (specific for Rmixmod results)
# It is better because probabilities of classification are generated
# by using the model parameters
resvisu <- clusvisMixmod(res)

# Component interpretation graph
plotDensityClusVisu(resvisu)

# Scatter-plot of the observation memberships
plotDensityClusVisu(resvisu, add.obs = TRUE)
```

## Second example: R package Rmixmod

```r
# Package loading
require(Rmixmod)

# Data loading (categorical data)
data(birds)

# Model-based clustering with 3 components
resmixmod <- mixmodCluster(birds, 3)

# Inference of the parameters used for results visualization (general approach)
# Probabilities of classification are not sampled from the model parameter,
```
clusvisVarSelLCM

This function estimates the parameters used for visualization of model-based clustering performs with R package Rmixmod. To achieve the parameter inference, it automatically samples probabilities of classification from the model parameters

Description

This function estimates the parameters used for visualization of model-based clustering performs with R package Rmixmod. To achieve the parameter inference, it automatically samples probabilities of classification from the model parameters

Usage

clusvisVarSelLCM(varselResult, sample.size = 5000, maxit = 1e+3, nbrandomInit = 4 * varselResult@model@g, nbcpu = 1, loccont = NULL)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>varselResult</td>
<td>[VSLCMresults] It is an instance of class VSLCMresults returned by function VarSelCluster of R package VarSelLCM.</td>
</tr>
<tr>
<td>sample.size</td>
<td>numeric. Number of probabilities of classification sampled for parameter inference.</td>
</tr>
<tr>
<td>maxit</td>
<td>numeric. It limits the number of iterations for the Quasi-Newton algorithm (default 1000).</td>
</tr>
<tr>
<td>nbrandomInit</td>
<td>numeric. It defines the number of random initialization of the Quasi-Newton algorithm.</td>
</tr>
<tr>
<td>nbcpu</td>
<td>numeric. It specifies the number of CPU (only for linux).</td>
</tr>
<tr>
<td>loccont</td>
<td>numeric. Index of the column containing continuous variables (only for mixed-type data).</td>
</tr>
</tbody>
</table>
**Value**

Returns a list

**Examples**

```r
## Not run:

# Package loading
require(VarSelLCM)

data("heart")

# Model-based clustering with 3 components
res <- VarSelCluster(heart[,13], 3)

# Inference of the parameters used for results visualization
# (specific for VarSelLCM results)
# It is better because probabilities of classification are generated
# by using the model parameters
resvisu <- clusvisVarSelLCM(res)

# Component interpretation graph
plotDensityClusVisu(resvisu)

# Scatter-plot of the observation memberships
plotDensityClusVisu(resvisu, add.obs = TRUE)

## End(Not run)
```

---

**congress**

*Real categorical data set: Congressional Voting Records Data Set*

**Description**

This data set includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the CQA. The CQA lists nine different types of votes: voted for, paired for, and announced for (these three simplified to yea), voted against, paired against, and announced against (these three simplified to nay), voted present, voted present to avoid conflict of interest, and did not vote or otherwise make a position known (these three simplified to an unknown disposition).

**References**


Website: https://archive.ics.uci.edu/ml/datasets/congressional+voting+records
plotDensityClusVisu

Function for visualizing the clustering results

Description

Function for visualizing the clustering results

Usage

plotDensityClusVisu(res, dim = c(1, 2), threshold = 0.95, add.obs = FALSE, positionlegend = "topright", xlim = NULL, ylim = NULL, colset = c("darkorange1", "dodgerblue2", "black", "chartreuse2", "darkorchid2", "gold2", "deeppink2", "deepskyblue1", "firebrick2", "cyan1", "red", "yellow"))

Arguments

- **res**: object return by function clusvis or clusvis
- **dim**: numeric. This vector of size two choose the axes to represent.
- **threshold**: numeric. It contains the thresholds used for computing the level curves.
- **add.obs**: boolean. If TRUE, coordinates of the observations are plotted.
- **positionlegend**: character. It specifies the legend location.
- **xlim**: numeric. It specifies the range of x-axis.
- **ylim**: numeric. It specifies the range of y-axis.
- **colset**: character. It specifies the colors of the observations per class.

Examples

```r
## Not run:
# Package loading
require(Rmixmod)

# Data loading (categorical data)
data("congress")
# Model-based clustering with 4 components
set.seed(123)
res <- mixmodCluster(congress[-1], 4, strategy = mixmodStrategy(nbTryInInit = 500, nbTry=25))

# Inference of the parameters used for results visualization
# (specific for Rmixmod results)
# It is better because probabilities of classification are generated
# by using the model parameters
resvisu <- clusvisMixmod(res)
```
# Component interpretation graph
plotDensityClusVisu(resvisu)

# Scatter-plot of the observation memberships
plotDensityClusVisu(resvisu, add.obs = TRUE)

## End(Not run)
Index

*Topic datasets
  congress, 9
*Topic package
  ClusVis-package, 2

ClusVis (ClusVis-package), 2
clusvis, 2, 4, 10
ClusVis-package, 2
clusvisMixmod, 2, 6
clusvisVarSelLCM, 2, 8
congress, 9

MixmodCluster, 6

plotDensityClusVisu, 2, 10

VSLCMresults, 8