Package ‘FPDclustering’

February 2, 2022

Type Package
Title PD-Clustering and Factor PD-Clustering
Version 2.1
Date 2022-01-31
Author Cristina Tortora [aut, cre, cph], Noe Viana [aut], Francesco Palumbo [aut], Tina Kalra [aut], and Paul D. McNicholas [fnd]
Maintainer Cristina Tortora <grikris1@gmail.com>
Description Probabilistic distance clustering (PD-clustering) is an iterative, distribution free, probabilistic clustering method. PD-clustering assigns units to a cluster according to their probability of membership, under the constraint that the product of the probability and the distance of each point to any cluster centre is a constant. PD-clustering is a flexible method that can be used with non-spherical clusters, outliers, or noisy data. PDQ is an extension of the algorithm for clusters of different size. GPDC and TPDC uses a dissimilarity measure based on densities. Factor PD-clustering (FPDC) is a factor clustering method that involves a linear transformation of variables and a cluster optimizing the PD-clustering criterion. It works on high dimensional data sets.
Depends ThreeWay, mvtnorm, R (>= 3.5)
Imports ExPosition, cluster, rootSolve, MASS, klaR, GGally, ggplot2
License GPL (>= 2)
NeedsCompilation no
Repository CRAN
Date/Publication 2022-02-02 09:00:04 UTC

R topics documented:

ais .......................................................... 2
asymmetric20 ............................................ 3
asymmetric3 ............................................. 4
FPDC .................................................... 4
GPDC .................................................... 6
outliers .................................................. 8
PDC ...................................................... 8
Description

Data obtained to study sex, sport and body-size dependency of hematology in highly trained athletes.

Usage

data(ais)

Format

A data frame with 202 observations and 13 variable.

rcc  red blood cell count, in
wcc  while blood cell count, in per liter
hc   hematocrit, percent
hg   hemaglobin concentration, in g per decaliter
ferr plasma ferritins, ng
bmi  Body mass index, kg
ssf  sum of skin folds
pcBfat percent Body fat
lcm  lean body mass, kg
ht   height, cm
wt   weight, kg
sex  a factor with levels f m
sport a factor with levels B_Ball Field Gym Netball Row Swim T_400m T_Sprnt Tennis W_Polo

Source

R package DAAG
References


Examples

data(ais)
pairs(ais[,1:11],col=ais$sex)

---

asymmetric20  Asymmetric data set shape 20

Description

Each cluster has been generated according to a multivariate asymmetric Gaussian distribution, with shape 20, covariance matrix equal to the identity matrix and randomly generated centres.

Usage

data(asymmetric20)

Format

A data frame with 800 observations on the following 101 variables. The first variable is the membership.

Source

Generated with R using the package sn (The skew-normal and skew-t distributions), function rsn

Examples

data(asymmetric20)
plot(asymmetric20[,2:3])
Asymmetric data set shape 3

Description
Each cluster has been generated according to a multivariate asymmetric Gaussian distribution, with shape 3, covariance matrix equal to the identity matrix and randomly generated centres.

Usage
data(asymmetric3)

Format
A data frame with 800 observations on 101 variables. The first variable is the membership labels.

Source
Generated with R using the package sn (The skew-normal and skew-t distributions), function rsn

Examples
data(asymmetric3)
plot(asymmetric3[,2:3])

---

Factor probabilistic distance clustering

Description
An implementation of FPDC, a probabilistic factor clustering algorithm that involves a linear transformation of variables and a cluster optimizing the PD-clustering criterion.

Usage
FPDC(data = NULL, k = 2, nf = 2, nu = 2)

Arguments
data A matrix or data frame such that rows correspond to observations and columns correspond to variables.
k A numerical parameter giving the number of clusters
nf A numerical parameter giving the number of factors for variables
nu A numerical parameter giving the number of factors for units
**FPDC**

**Value**

A class FPDclustering list with components

- **label**: A vector of integers indicating the cluster membership for each unit
- **centers**: A matrix of cluster centers
- **probability**: A matrix of probability of each point belonging to each cluster
- **JDF**: The value of the Joint distance function
- **iter**: The number of iterations
- **explained**: The explained variability
- **data**: the data set

**Author(s)**

Cristina Tortora and Paul D. McNicholas

**References**


**See Also**

PDC

**Examples**

```r
## Not run:
# Asymmetric data set clustering example (with shape 3).
data('asymmetric3')
x<-asymmetric3[,,-1]

#Clustering
fpdas3=FPDC(x,4,3,3)

#Results
table(asymmetric3[,1],fpdas3$label)
Silh(fpdas3$probability)
summary(fpdas3)
plot(fpdas3)

## End(Not run)
```
# Asymmetric data set clustering example (with shape 20).
data('asymmetric20')
x<-asymmetric20[,,-1]

#Clustering
fpdas20=FPDC(x,4,3,3)

#Results
table(asymmetric20[,1],fpdas20$label)
Silh(fpdas20$probability)
summary(fpdas20)
plot(fpdas20)

## End(Not run)

## Not run:
# Clustering example with outliers.
data('outliers')
x<-outliers[,,-1]

#Clustering
fpdout=FPDC(x,4,5,4)

#Results
table(outliers[,1],fpdout$label)
Silh(fpdout$probability)
summary(fpdout)
plot(fpdout)

## End(Not run)

---

**GPDC**

*Gaussian PD-Clustering*

**Description**

An implementation of Gaussian PD-Clustering GPDC, an extension of PD-clustering adjusted for cluster size that uses a dissimilarity measure based on the Gaussian density.

**Usage**

GPDC(data=NULL,k=2,method="kmedoids", nr=5,iter=100)

**Arguments**

- **data** A matrix or data frame such that rows correspond to observations and columns correspond to variables.
- **k** A numerical parameter giving the number of clusters
method A parameter that selects center starts. Options available are random ("random"), kmedoid ("kmedoid", by default), and PDC ("PDclust").

nr Number of random starts when method set to "random"

iter Maximum number of iterations

Value

A class FPDclustering list with components

label A vector of integers indicating the cluster membership for each unit
centers A matrix of cluster means
sigma A list of K elements, with the variance-covariance matrix per cluster
probability A matrix of probability of each point belonging to each cluster
JDF The value of the Joint distance function
iter The number of iterations
data the data set

Author(s)

Cristina Tortora and Francesco Palumbo

References


See Also

PDC, PDQ

Examples

#Load the data
data(ais)
dataSEL=ais[,c(10,3,5,8)]

#Clustering
res=GPDC(dataSEL,k=2,method = "kmedoids")

#Results
table(res$label,ais$sex)
plot(res)
summary(res)
PDC

**Description**

Each cluster has been generated according to a multivariate Gaussian distribution, with centers $c$ randomly generated. For each cluster, 20% of uniform distributed outliers have been generated at a distance included in $\text{max}(x-c)$ and $\text{max}(x-c)+5$ form the center.

**Usage**

```r
data(outliers)
```

**Format**

A data frame with 960 observations on the following 101 variables. The first variable corresponds to the membership

**Source**

generated with R

**Examples**

```r
data(outliers)
plot(outliers[,2:3])
```

---

**PDC**

*Probabilistic Distance Clustering*

**Description**

Probabilistic distance clustering (PD-clustering) is an iterative, distribution free, probabilistic clustering method. PD clustering is based on the constraint that the product of the probability and the distance of each point to any cluster centre is a constant.

**Usage**

```r
PDC(data = NULL, k = 2)
```

**Arguments**

- `data` A matrix or data frame such that rows correspond to observations and columns correspond to variables.
- `k` A numerical parameter giving the number of clusters
Value

A class FPDclustering list with components

- **label**: A vector of integers indicating the cluster membership for each unit
- **centers**: A matrix of cluster centers
- **probability**: A matrix of probability of each point belonging to each cluster
- **JDF**: The value of the Joint distance function
- **iter**: The number of iterations
- **data**: The data set

Author(s)

Cristina Tortora and Paul D. McNicholas

References


Examples

```r
# Normally generated clusters
c1 = c(+2,+2,2,2)
c2 = c(-2,-2,-2,-2)
c3 = c(-3,3,-3,3)
n=200
x1 = cbind(rnorm(n, c1[1]), rnorm(n, c1[2]), rnorm(n, c1[3]), rnorm(n, c1[4]) )
x2 = cbind(rnorm(n, c2[1]), rnorm(n, c2[2]),rnorm(n, c2[3]), rnorm(n, c2[4]) )
x3 = cbind(rnorm(n, c3[1]), rnorm(n, c3[2]),rnorm(n, c3[3]), rnorm(n, c3[4]) )
x = rbind(x1,x2,x3)

# Clustering
pdn=PDC(x,3)

# Results
plot(pdn)
```

Probabilistic Distance Clustering Adjusted for Cluster Size

Description

An implementation of probabilistic distance clustering adjusted for cluster size (PDQ), a probabilistic distance clustering algorithm that involves optimizing the PD-clustering criterion. The algorithm can be used, on continuous, count, or mixed type data setting Euclidean, Chi square, or Gower as dissimilarity measurements.
Usage

PDQ(x=NULL,k=2,ini='kmd',dist='euc',cent=NULL,ord=NULL,cat=NULL,bin=NULL,cont=NULL,w=NULL)

Arguments

**x**  
A matrix or data frame such that rows correspond to observations and columns correspond to variables.

**k**  
A numerical parameter giving the number of clusters.

**ini**  
A parameter that selects center starts. Options available are random ("random"), kmedoid ("kmd", by default), center ("center", the user inputs the center), and kmode ("kmode", for categoriacal data sets).

**dist**  
A parameter that selects the distance measure used. Options available are Euclidean ("euc"), Gower ("gower") and chi square ("chi").

**cent**  
User inputed centers if method selected is "random".

**ord**  
column indices of the x matrix indicating which columns are ordinal variables.

**cat**  
column indices of the x matrix indicating which columns are categorical variables.

**bin**  
column indices of the x matrix indicating which columns are binary variables.

**cont**  
column indices of the x matrix indicating which columns are continuous variables.

**w**  
umerical vector same length as the columns of the data, containing the variable weights when using Gower distance, equal weights by default.

Value

A class FPDclustering list with components

**label**  
A vector of integers indicating the cluster membership for each unit

**centers**  
A matrix of cluster centers

**probability**  
A matrix of probability of each point belonging to each cluster

**JDF**  
The value of the Joint distance function

**iter**  
The number of iterations

**jdfvector**  
collection of all jdf calculations at each iteration

**data**  
the data set

Author(s)

Cristina Tortora and Noe Vidales

References


See Also

PDC

Examples

# Mixed type data

```r
sig = matrix(0.7, 4, 4)
diag(sig) = 1## creat a correlation matrix
x1 = rmvnorm(200, c(0, 0, 3, 3))## cluster 1
x2 = rmvnorm(200, c(4, 4, 6, 6), sigma = sig)## cluster 2
x = rbind(x1, x2)# data set with 2 clusters
l = c(rep(1, 200), rep(2, 200))# creating the labels
x1 = cbind(x1, rbinom(200, 4, 0.2), rbinom(200, 4, 0.2))# categorical variables
x2 = cbind(x2, rbinom(200, 4, 0.7), rbinom(200, 4, 0.7))
x = rbind(x1, x2)## Data set

### Performing PDQ

pdq_class <- PDQ(x = x, k = 2, ini = "random", dist = "gower", cont = 1:4, cat = 5:6)

### Output

table(l, pdq_class$label)
plot(pdq_class)
summary(pdq_class)
```

### Continuous data example

# Gaussian Generated Data no overlap

```r
x <- rmvnorm(100, mean = c(1, 5, 10), sigma = diag(1, 3))
y <- rmvnorm(100, mean = c(4, 8, 13), sigma = diag(1, 3))
data <- rbind(x, y)

### Performing PDQ

pdq1 = PDQ(data, 2, ini = "random", dist = "euc")
table(rep(c(2, 1), each = 100), pdq1$label)
Silh(pdq1$probability)
plot(pdq1)
summary(pdq1)
```

# Gaussian Generated Data with overlap

```r
x2 <- rmvnorm(100, mean = c(1, 5, 10), sigma = diag(1, 3))
y2 <- rmvnorm(100, mean = c(2, 6, 11), sigma = diag(1, 3))
data2 <- rbind(x2, y2)

### Performing PDQ

pdq2 = PDQ(data2, 2, ini = "random", dist = "euc")
table(rep(c(1, 2), each = 100), pdq2$label)
plot(pdq2)
summary(pdq2)
```
Description

Probability Silhouette plot, Scatterplot up to 10 variables, and parallel coordinate plot up to 10 variables, for objects of class FPDclustering.

Usage

## S3 method for class 'FPDclustering'
plot(x, maxVar=30, ...)

Arguments

x an object of class FPDclustering
maxVar a scalar indicating the maximum number of variables to display on the parallel plot, 30 by default
... Additional parameters for the function paris

Author(s)

Cristina Tortora

Silh

Probabilistic silhouette plot

Description

Graphical tool to evaluate the clustering partition.

Usage

Silh(p)

Arguments

p A matrix of probabilities such that rows correspond to observations and columns correspond to clusters.
Details

The probabilistic silhouettes are an adaptation of the ones proposed by Menardi(2011) according to the following formula:

\[
dbs_i = \frac{\log(p_{ik}/p_{im_1})}{\max_i \{|\log(p_{ik}/p_{im})|\}}
\]

where \(m_k\) is such that \(x_i\) belongs to cluster \(k\) and \(m_1\) is such that \(p_{im_1}\) is maximum for \(m\) different from \(m_k\).

Value

Probabilistic silhouette plot

Author(s)

Cristina Tortora

References


Examples

```r
## Not run:
# Asymmetric data set silhouette example (with shape=3).
data("asymmetric3")
x<-asymmetric3[,,-1]
fpdas3=FPDC(x,4,3,3)
Silh(fpdas3$probability)
## End(Not run)

## Not run:
# Asymmetric data set shiluette example (with shape=20).
data("asymmetric20")
x<-asymmetric20[,,-1]
fpdas20=FPDC(x,4,3,3)
Silh(fpdas20$probability)
## End(Not run)

## Not run:
# Shiluette example with outliers.
data("outliers")
x<-outliers[,,-1]
fpdout=FPDC(x,4,4,3)
Silh(fpdout$probability)
## End(Not run)
```
### Star

*Star dataset to predict star types*

#### Description

A 6 class star dataset for star classification with Deep Learned approaches.

#### Usage

```r
data(ais)
```

#### Format

A data frame with 202 observations and 13 variables.

- **K**: Absolute Temperature (in K)
- **Lum**: Relative Luminosity (L/Lo)
- **Rad**: Relative Radius (R/Ro)
- **Mag**: Absolute Magnitude (Mv)
- **Col**: Star Color (white, Red, Blue, Yellow, yellow-orange etc)
- **Spect**: Spectral Class (O, B, A, F, G, K, M)
- **Type**: Star Type (Red Dwarf, Brown Dwarf, White Dwarf, Main Sequence, SuperGiants, Hyper-Giants)

#### Source

https://www.kaggle.com/deepu1109/star-dataset

#### Examples

```r
data(Star)
```

### summary.FPDclustering

*Summary for FPDclusteringt Objects*

#### Description

Number of elements per cluster.

#### Usage

```r
## S3 method for class 'FPDclustering'
summary(object, ... )
```
Arguments

- object: an object of class FPclustering
- Additional parameters for the function pari

Author(s)

Cristina Tortora

Description

An implementation of Student-t PD-Clustering TPDC, an extension of PD-clustering adjusted for cluster size that uses a dissimilarity measure based on the multivariate Student-t density.

Usage

TPDC(data=NULL, k=2, method="kmedoids", nr=5, iter=100)

Arguments

- data: A matrix or data frame such that rows correspond to observations and columns correspond to variables.
- k: A numerical parameter giving the number of clusters.
- method: A parameter that selects center starts. Options available are random ("random"), kmedoid ("kmedoid", by default), and PDC ("PDclust").
- nr: Number of random starts if method is "random"
- iter: Maximum number of iterations

Value

A class FPclustering list with components

- label: A vector of integers indicating the cluster membership for each unit
- centers: A matrix of cluster means
- sigma: A list of K elements, with the variance-covariance matrix per cluster
- df: A vector of K degrees of freedom
- probability: A matrix of probability of each point belonging to each cluster
- JDF: The value of the Joint distance function
- iter: The number of iterations
- data: the data set
Author(s)
Cristina Tortora and Francesco Palumbo

References

See Also
PDC, PDQ

Examples
#Load the data
data(ais)
dataSEL=ais[,c(10,3,5,8)]

#Clustering
res=TPDC(dataSEL,k=2,method = "kmedoids")

#Results
table(res$label,ais$sex)
summary(res)
plot(res)

TuckerFactors

Choice of the number of Tucker 3 factors for FPDC

Description
An empirical way of choosing the number of factors for FPDC. The function returns a graph and a table representing the explained variability varying the number of factors.

Usage
TuckerFactors(data = NULL, nc = 2)

Arguments
data A matrix or data frame such that rows correspond to observations and columns correspond to variables.
nc A numerical parameter giving the number of clusters
**Value**

A table containing the explained variability varying the number of factors for units (column) and for variables (row) and the corresponding plot.

**Author(s)**

Cristina Tortora

**References**


**See Also**

T3

**Examples**

```r
## Not run:
# Asymmetric data set example (with shape=3).
data("asymmetric3")
xp=TuckerFactors(asymmetric3[,-1], nc = 4)

## End(Not run)

## Not run:
# Asymmetric data set example (with shape=20).
data("asymmetric20")
xp=TuckerFactors(asymmetric20[,-1], nc = 4)

## End(Not run)
```
Index

ais, 2
asymmetric20, 3
asymmetric3, 4

FPDC, 4

GPDC, 6

outliers, 8

PDC, 5, 7, 8, 11, 16
PDQ, 7, 9, 16
plot.FPDclustering, 12

Silh, 12
Star, 14
summary.FPDclustering, 14

T3, 17
TPDC, 15
TuckerFactors, 16