Package 'RJcluster'

February 14, 2022

Title A Fast Clustering Algorithm for High Dimensional Data Based on the Gram Matrix Decomposition

Version 3.2.4

Author Shahina Rahman [aut],
Valen E. Johnson [aut],
Suhasini Subba Rao [aut],
Rachael Shudde [aut, cre, trl]

Maintainer Rachael Shudde <rachael.shudde@gmail.com>

Description Clustering algorithm for high dimensional data. Assuming that P feature measurements on N objects are arranged in an N×P matrix X, this package provides clustering based on the left Gram matrix XX^T. To simulate test data, type ``help('simulate_HD_data')'' and to learn how to use the clustering algorithm, type ``help('RJclust')''. To cite this package, type 'citation(``RJcluster'')'.

License GPL (>= 2)

Encoding UTF-8

Imports Rcpp (>= 1.0.2), matrixStats, infotheo, rlang, stats, graphics, profvis, mclust, foreach, utils

LinkingTo Rcpp, RcppArmadillo

Suggests testthat (>= 2.1.0), knitr, rmarkdown

RoxygenNote 7.1.1

VignetteBuilder knitr

Depends R (>= 2.10)

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-02-14 21:30:02 UTC

R topics documented:

RJcluster-package 2
Mutual_Information 2
RJclust 3
simulate_HD_data 5

Index **7**

`RJcluster-package` *A Fast Clustering Algorithm for High Dimensional Data Based on the Gram Matrix Decomposition*

Description

Clustering algorithm for high dimensional data. Assuming that P feature measurements on N objects are arranged in an N×P matrix X, this package provides clustering based on the left Gram matrix XX^T. To simulate test data, type "help('simulate_HD_data')" and to learn how to use the clustering algorithm, type "help('RJclust')". To cite this package, type 'citation("RJcluster")'.

Details

Package:	RJcluster
Type:	Package
Version:	3.2.4
Date:	07-15-2021
License:	GPL>=2

Author(s)

Shahina Rahman [aut], Valen E. Johnson [aut], Suhasini Subba Rao [aut], Rachael Shudde [aut, cre, trl]

Maintainer: Rachael Shudde <rachael.shudde@gmail.com>

`Mutual_Information` *Mutual_Information*

Description

Calculates normalized mutual information and adjusted mutual information. The value for both will be a value bewteen 0 and 1 that measures how close the classification between the two clusters is. A value closer to 1 means the labels are more similar across v1 and v2, and a value closer to 0 means the labels are not as similar.

Usage

```
Mutual_Information(v1, v2)
```

Arguments

`v1`	vector containing first classification labels
`v2`	vector containing second classification labels

Details

See these links for a more formal definition of AMI and NMI.

Value

Returns mutual information:

`nmi`	NMI value
`ami`	AMI value

Examples

```
cluster1 <- sample(1:5, size = 10, replace = TRUE)
cluster2 <- sample(1:2, size = 10, replace = TRUE)
Mutual_Information(cluster1, cluster2)
```

`RJclust` *RJclust*

Description

This is a high dimensional clustering algorithm for data in matrix form. There are are two different types of penalty methods that can be used, depending on the size of the data and the desired accuracy. The first is the default method: the hokey stick penalty. There is also the BIC penalty. For large n, the scale method can be used, which uses the approximation method of RJclust. For the scaleRJ method, a parmater n_bins (usually $\sqrt(p)$) is required that splits the data into different buckets. For all methods, a C_max variable is needed that is an upper limit on the possible number of clusters.

Usage

```
RJclust(
  data,
  penalty = "hockey_stick",
  scaleRJ = FALSE,
  C_max = 10,
  criterion = "VVI",
  n_bins = NULL,
  seed = 1,
  verbose = FALSE
)
```

Arguments

`data`	Data input, must be in matrix form. Currently no support for missing values
`penalty`	A string of possible vectors. Options include: "bic" an "hockey_stock" (default = "hockey_stick")
`scaleRJ`	Should the scaled version of RJ be used, suggested for data where n > 1000 (default = FALSE)
`C_max`	Maximum number of clusters to look for (default is 10)
`criterion`	Model of covariance structure (default = "VVI")
`n_bins`	Number of cuts if penalty = "scale" for the scaled RJ algorithm (default = sqrt(p))
`seed`	Seed (defalt = 1)
`verbose`	Should progress be printed? (default = FALSE)

Details

All implementations use backend C++ to increase runtime.

model_names controls the type of covariance structure. See Mclust Documenttion for more information. Note criterion "kmeans" is the same as "EEI". It is not suggested to use "kmeans" if it is suspected the classes are imbalanced

Value

Returns RJ algorithm result for "aic", "bic" ("mclust" and "scale" will return an mclust object:

`K`	number of clusters found
`class`	Class labels
`penalty`	Penalty values at each iteraiton
`mean`	Mean matrix
`prob`	Probability values
`z`	Z values from mclust (NULL penalty = "full_covariance")

Examples

```
X = simulate_HD_data()
X = X$X
clust = RJclust(X, penalty = "hockey_stick", C_max = 10)
```

`simulate_HD_data` *simulate_HD_data*

Description

This is simulaiton data to check performance of RJcluster. Data can be simulated for any n, P, and size of clusters. The data has two types of data: noisy data and signal data. The percent of the data that is noisy is controlled by the sparsity paramater. The noisy data has two parts: half of it is $N(0, 1)$ and half is $N(0, noise_variance)$. The signal data is divided in two as well, half of it is $N(\mu[, 1], signal_variance)$ and half $N(\mu[, 2], signal_variance)$.

Usage

```
simulate_HD_data(
  size_vector = c(20, 20, 20, 20),
  p = 220,
  mu = matrix(c(1.5, 2.5, 0, 1.5, 0, -1.5, -2.5, -1.5), ncol = 2, byrow = TRUE),
  signal_variance = 1,
  noise_variance = 1,
  sparsity = 0.09,
  seed = 1234
)
```

Arguments

`size_vector`	A list of the size of the different clusters. (default = a balanced case of 4 clusters of size 20, c(20, 20, 20, 20))
`p`	The number of columns in the simulated matrix (default = 220)
`mu`	The matrix of means, of dimension length(size_vector)x2. The first column of means is for the first half informative features, the second columns of mean is for the second half of the informative features (default is described in RJcluster paper)
`signal_variance`	Variance of the signal part of the generated data. A value of 1 indicates a high SNR, a value of 2 indicates a low SNR (default = 1)
`noise_variance`	Variance of the noisy part of the generated data (Default = 1)
`sparsity`	What percent of the data should be informative? A value between 0 and 1, a higher value means more data is informative (default = 0.09)
`seed`	Random seed. Change if generating multiple simulation datasets (default = 1234)

Details

The data in the paper is generated with number of clusters = 4, a balanced case of c(20, 20, 20, 20) and an unbalanced case of c(20, 20, 200, 200), with p = 220 in both cases. The default is a balanced, high signal case with μ as the matrix in the RJcluster paper.

Value

Returns simulation data for X and Y values

X Matrix of dimension sum(size_vector)xp

Y Vector of class labels of length $\sum(size_vector)$, with unique values of 1:length(size_vector)

Examples

```
data = simulate_HD_data()
X = data$X
Y = data$X
print(head(X))
```

Index

Mutual_Information, 2

RJclust, 3
RJcluster *(*RJcluster-package*)*, 2
RJcluster-package, 2

simulate_HD_data, 5