Package ‘QuClu’

October 12, 2022

Type Package
Title Quantile-Based Clustering Algorithms
Version 1.0.1
Date 2022-05-26
Author Christian Hennig, Cinzia Viroli and Laura Anderlucci
Maintainer Laura Anderlucci <laura.anderlucci@unibo.it>
License GPL-2 | GPL-3
Encoding UTF-8
RoxygenNote 7.2.0
Imports stats
NeedsCompilation no
Repository CRAN
Date/Publication 2022-05-26 16:40:02 UTC

R topics documented:

alg.CS .................................................. 2
alg.CU .................................................. 3
alg.VS .................................................. 4
alg.VU .................................................. 5
kquantiles .......................................... 7

Index 9
**alg.CS**

*CS quantile-based clustering algorithm*

**Description**

This function allows to run the CS (Common theta and Scaled variables through lambda_j) version of the quantile-based clustering algorithm.

**Usage**

```
alg.CS(data, k = 2, eps = 1e-08, it.max = 100, B = 30, lambda = rep(1, p))
```

**Arguments**

- `data` A numeric vector, matrix, or data frame of observations. Categorical variables are not allowed. If a matrix or data frame, rows correspond to observations and columns correspond to variables.
- `k` The number of clusters. The default is k=2.
- `eps` The relative convergence tolerances for objective function. The default is set to 1e-8.
- `it.max` A number that gives integer limits on the number of the CS algorithm iterations. By default, it is set to 100.
- `B` The number of times the initialization step is repeated; the default is 30.
- `lambda` The initial value for lambda_j, the variable scaling parameters. By default, lambdas are set to be equal to 1.

**Details**

Algorithm CS: Common theta and Scaled variables via lambda_j. A common value of theta is taken but variables are scaled through lambda_j.

**Value**

A list containing the following elements:

- `cl` A vector whose [i]th entry is classification of observation i in the test data.
- `qq` A matrix whose [h,j]th entry is the theta-quantile of variable j in cluster h.
- `theta` The estimated common theta.
- `Vseq` The values of the objective function V at each step of the algorithm.
- `V` The final value of the objective function V.
- `lambda` A vector containing the scaling factor for each variable.

**References**

Examples

```r
out <- alg.CS(iris[, -5], k = 3)
out$theta
out$qq
out$lambda

table(out$cl)
```

---

**alg.CU**

CU quantile-based clustering algorithm

---

**Description**

This function allows to run the CU (Common theta and Unscaled variables) version of the quantile-based clustering algorithm.

**Usage**

```r
alg.CU(data, k = 2, eps = 1e-08, it.max = 100, B = 30)
```

**Arguments**

- `data` A numeric vector, matrix, or data frame of observations. Categorical variables are not allowed. If a matrix or data frame, rows correspond to observations and columns correspond to variables.
- `k` The number of clusters. The default is `k=2`.
- `eps` The relative convergence tolerances for objective function. The default is set to `1e-8`.
- `it.max` A number that gives integer limits on the number of the CU algorithm iterations. By default, it is set to `100`.
- `B` The number of times the initialization step is repeated; the default is `30`.

**Details**

Algorithm CU: Common theta and Unscaled variables. A common value of theta for all the variables is assumed. This strategy directly generalizes the conventional k-means to other moments of the distribution to better accommodate skewness in the data.

**Value**

A list containing the following elements:

- `method` The chosen parameterization, CU, Common theta and Unscaled variables
- `k` The number of clusters.
- `cl` A vector whose [i]th entry is classification of observation i in the test data.
- `qq` A matrix whose [h,j]th entry is the theta-quantile of variable j in cluster h.
theta A vector whose \( j \)th entry is the percentile theta for variable \( j \).

\( \text{Vseq} \) The values of the objective function \( V \) at each step of the algorithm.

\( V \) The final value of the objective function \( V \).

\( \lambda \) A vector containing the scaling factor for each variable.

References


Examples

```r
out <- alg.CU(iris[,,-5],k=3)
out$theta
out$qq

table(out$cl)
```

```
alg.VS VS quantile-based clustering algorithm
```

Description

This function allows to run the VS (Variable-wise theta \( j \) and Scaled variables through \( \lambda \)) version of the quantile-based clustering algorithm.

Usage

```r
alg.VS(data, k = 2, eps = 1e-08, it.max = 100, B = 30, lambda = rep(1, p))
```

Arguments

data A numeric vector, matrix, or data frame of observations. Categorical variables are not allowed. If a matrix or data frame, rows correspond to observations and columns correspond to variables.

k The number of clusters. The default is k=2.

eps The relative convergence tolerances for objective function. The default is set to 1e-8.

it.max A number that gives integer limits on the number of the VS algorithm iterations. By default, it is set to 100.

B The number of times the initialization step is repeated; the default is 30.

lambda The initial value for lambda \( j \), the variable scaling parameters. By default, lambdas are set to be equal to 1.
**Details**

Algorithm VS: Variable-wise theta_j and Scaled variables via lambda_j. A different theta for every single variable is estimated to better accommodate different degree of skeweness in the data and variables are scaled through lambda_j.

**Value**

A list containing the following elements:

- `method`: The chosen parameterization, VS, Variable-wise theta_j and Scaled variables
- `k`: The number of clusters.
- `cl`: A vector whose [i]th entry is classification of observation i in the test data.
- `qq`: A matrix whose [h,j]th entry is the theta-quantile of variable j in cluster h.
- `theta`: A vector whose [j]th entry is the percentile theta for variable j.
- `Vseq`: The values of the objective function V at each step of the algorithm.
- `V`: The final value of the objective function V.
- `lambda`: A vector containing the scaling factor for each variable.

**References**


**Examples**

```r
out <- alg.VS(iris[-5], k=3)
out$theta
out$qq
out$lambda

table(out$cl)
```

---

**alg.VU**

*VU quantile-based clustering algorithm*

**Description**

This function allows to run the VU (Variable-wise theta_j and Unscaled variables) version of the quantile-based clustering algorithm.

**Usage**

```r
alg.VU(data, k = 2, eps = 1e-08, it.max = 100, B = 30)
```
Arguments

- **data**: A numeric vector, matrix, or data frame of observations. Categorical variables are not allowed. If a matrix or data frame, rows correspond to observations and columns correspond to variables.

- **k**: The number of clusters. The default is k=2.

- **eps**: The relative convergence tolerances for objective function. The default is set to 1e-8.

- **it.max**: A number that gives integer limits on the number of the VU algorithm iterations. By default, it is set to 100.

- **B**: The number of times the initialization step is repeated; the default is 30.

Details

Algorithm VU: Variable-wise theta_j and Unscaled variables. A different theta for every single variable is estimated to better accommodate different degree of skewness in the data.

Value

A list containing the following elements:

- **method**: The chosen parameterization, VU, Variable-wise theta_j and Unscaled variables
- **k**: The number of clusters.
- **cl**: A vector whose [i]th entry is classification of observation i in the test data.
- **qq**: A matrix whose [h,j]th entry is the theta-quantile of variable j in cluster h.
- **theta**: A vector whose [j]th entry is the percentile theta for variable j.
- **Vseq**: The values of the objective function V at each step of the algorithm.
- **V**: The final value of the objective function V.
- **lambda**: A vector containing the scaling factor for each variable.

References


Examples

```r
out <- alg.VU(iris[-5], k=3)
out$theta
out$qq

table(out$cl)
```
**kquantiles**  

**Quantile-based clustering algorithm**

**Description**

This function allows to run the $k$-quantile clustering algorithm, allowing for different constraints: common theta and unscaled variables (CU), common theta and scaled variables (CS), variable-wise theta and unscaled variables (VU) and the variable-wise theta and scaled variables (VS).

**Usage**

```r
kquantiles(
  data,
  k = 2,
  method = "VS",
  eps = 1e-08,
  it.max = 100,
  B = 30,
  lambda = NULL
)
```

**Arguments**

- **data**  
  A numeric vector, matrix, or data frame of observations. Categorical variables are not allowed. If a matrix or data frame, rows correspond to observations and columns correspond to variables.

- **k**  
  The number of clusters. The default is $k=2$.

- **method**  
  The chosen constrained method. The options are: CU (Common theta and Unscaled variables), CS (Common theta and Scaled variables), VU (Variable-wise theta and Unscaled variables), VS (Variable-wise theta and Scaled variables). The default is the unconstrained method, VS.

- **eps**  
  The relative convergence tolerances for objective function. The default is set to $1e-8$.

- **it.max**  
  A number that gives integer limits on the number of the algorithm iterations. By default, it is set to 100.

- **B**  
  The number of times the initialization step is repeated; the default is 30.

- **lambda**  
  The initial value for lambda_j, the variable scaling parameters, for models CS and VS. By default, lambdas are set to be equal to 1.

**Details**

Algorithm CU: Common theta and Unscaled variables. A common value of theta for all the variables is assumed. Algorithm CS: Common theta and Scaled variables via lambda_j. A common value of theta is taken but variables are scaled through lambda_j. Algorithm VU: Variable-wise theta_j and Unscaled variables. A different theta for every single variable is estimated to better
accomodate different degree of skeweness in the data. Algorithm VS: Variable-wise theta_j and Scaled variables via lambda_j. A different theta for every single variable is estimated to better accommodate different degree of skeweness in the data and variables are scaled through lambda_j.

Value
A list containing the following elements:

- **method**: The chosen parameterization.
- **k**: The number of clusters.
- **cl**: A vector whose [i]th entry is classification of observation i in the test data.
- **qq**: A matrix whose [h,j]th entry is the theta-quantile of variable j in cluster h.
- **theta**: A vector whose [j]th entry is the percentile theta for variable j.
- **Vseq**: The values of the objective function V at each step of the algorithm.
- **V**: The final value of the objective function V.
- **lambda**: A vector containing the scaling factor for each variable.

References

Examples
```r
out <- kquantiles(iris[, -5], k = 3, method = "VS")
out$theta
out$qq

table(out$cl)
```
Index

alg. CS, 2
alg. CU, 3
alg. VS, 4
alg. VU, 5

kquantiles, 7