Package ‘GBClust’

November 27, 2020

Type Package

Title Clustering with Gibbs posteriors

Version 0.0.1

Date 2020-11-25

Author Tommaso Rigon

Maintainer Tommaso Rigon <tommaso.rigon@unimib.it>

Description This package is an implementation of the generalized Bayes clustering methods described in the paper by Rigon, T., Herring, A. H. and Dunson, D. B. (2020), entitled "A generalized Bayes framework for probabilistic clustering”

Encoding UTF-8

License GPL-3

LazyData TRUE

Imports Rcpp (>= 1.0.2), ggplot2, cluster

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.1.1

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

comp_medoids *Computation of the medoids*

Description

Compute the medoids of a given clustering solution based on the corresponding dissimilarity matrix.

Usage

```
comp_medoids(D, cluster)
```

Arguments

`D`	A `n x n` numeric matrix with the dissimilarities, usually the output of `dist` or `daisy`.
`cluster`	A clustering solution, typically the output of `kdiss`.

Value

`medoids` Indexes of the medoids following the order of the dissimilarity matrix `D`.

kbinary *K-binary clustering*

Description

Perform k-binary clustering

Usage

```
kbinary(x, k, nstart = 1, trace = FALSE)
```

Arguments

`x`	numeric matrix of data, or an object that can be coerced to such a matrix (such as a numeric vector or a data frame with all numeric columns).
`k`	The number of clusters to be considered. A random set of (distinct) rows in x is chosen as the initial centres.
`nstart`	Number of random sets that has been chosen
`trace`	logical: if true, tracing information on the progress of the algorithm is produced.

Value

- A - The letters of the alphabet.
- B - A vector of numbers.

`kbinary_gibbs` *K-dissimilarities algorithm with uncertainty quantification*

Description

Perform the Gibbs-sampling for the k-dissimilarities algorithm using the Minkowski distance; see `dist`.

Usage

```
kbinary_gibbs(
  x,
  k,
  lambda = 1,
  R = 1000,
  burn_in = 1000,
  nstart = 10,
  trace = FALSE
)
```

Arguments

`x`	numeric matrix of of the data
`k`	The number of clusters to be considered.
`R`	Number of MCMC samples after burn-in
`burn_in`	Number of MCMC samples to be discarded as burn-in period
`nstart`	Number of random initializations for the k-means algorithm
`trace`	logical: if true, tracing information on the progress of the algorithm is produced.

Value

`G` The letters of the alphabet

`lambda` A vector of numbers

`loss` A vector of numbers

`G_map` A vector of numbers

`loss_map` A vector of numbers

`kbinary_select` *Selection of the number of cluster for the k-binary algorithm*

Description

It displays the value of the loss function for various choices of k

Usage

```
kbinary_select(x, k_max, nstart = 1)
```

Arguments

`x`	numeric matrix of data, or an object that can be coerced to such a matrix (such as a numeric vector or a data frame with all numeric columns).
`k_max`	The maximum number of clusters to be considered. A random set of (distinct) rows in x is chosen as the initial centres.
`nstart`	Number of random sets that has been chosen

Value

It plots the loss function for different clustering solutions

---

`kdiss` *K-dissimilarities algorithm*

Description

Perform the k-dissimilarities algorithm described in Rigon, T., Herring, A. H. and Dunson, D. B. (2020).

Usage

```
kdiss(D, k, nstart = 1, trace = FALSE)
```

Arguments

`D`	A `n x n` numeric matrix with the dissimilarities, typically the output of `dist` or `daisy`.
`k`	The number of clusters to be considered. See `kdiss_select` for selection criteria.
`nstart`	Number of random initializations.
`trace`	logical: if true, tracing information on the progress of the algorithm is produced

Value

`cluster` Labels of the clusters at convergence

`loss` Numeric value of the loss function at convergence

`kdiss_select` *Selection of the number of cluster for the k-dissimilarities algorithm*

Description

It displays the value of the loss function / average silhouette width, for different values of `k`

Usage

```
kdiss_select(D, k_max, nstart = 1, method = "elbow")
```

Arguments

`D`	A `n x n` numeric matrix with the dissimilarities, typically the output of `dist` or `daisy`.
`k_max`	Maximum number of clusters to be considered.
`nstart`	Number of random initializations.
`method`	The graph that will be displayed. Supported options are `method="elbow"`, which displays the loss function, or `method="silhouette"`. See `silhouette` for details about the latter.

Value

It return a `ggplot2` graph of the loss function / average silhouette width, for `k=1,...,k_max`.

`kmeans2` *K-Means^2 Clustering*

Description

Perform k-means and k-means^2 on a data matrix

Usage

```
kmeans2(x, k, nstart = 1, algorithm = "kmeans", trace = FALSE)
```

Arguments

`x`	numeric matrix of data, or an object that can be coerced to such a matrix (such as a numeric vector or a data frame with all numeric columns).
`k`	The number of clusters to be considered. A random set of (distinct) rows in x is chosen as the initial centres.
`nstart`	Number of random sets that has been chosen
`algorithm`	The algorithm to be used
`trace`	logical: if true, tracing information on the progress of the algorithm is produced.

Value

- A - The letters of the alphabet.
- B - A vector of numbers.

kmeans2_select *Selection of the number of cluster for the k-dissimilarities algorithm*

Description

It displays the value of the loss function for various choices of k

Usage

```
kmeans2_select(x, k_max, nstart = 1, algorithm = "kmeans")
```

Arguments

`x`	numeric matrix of data, or an object that can be coerced to such a matrix (such as a numeric vector or a data frame with all numeric columns).
`k_max`	The maximum number of clusters to be considered. A random set of (distinct) rows in x is chosen as the initial centres.
`nstart`	Number of random sets that has been chosen
`algorithm`	The algorithm to be used, either `kmeans` or `kmeans2`

Value

It plots the loss function for different clustering solutions

kmeans_gibbs *K-means clustering with uncertainty quantification*

Description

Perform the Gibbs-sampling for the k-means algorithm, as described in Rigon, Herring and Dunson (2020).

Usage

```
kmeans_gibbs(
  x,
  k,
  a_lambda,
  b_lambda,
  R = 1000,
  burn_in = 1000,
  nstart = 10,
  trace = FALSE
)
```

Arguments

`x`	A `n x d` numeric matrix of the data.
`k`	The number of clusters to be considered.
`a_lambda`	Hyperparameter of the Gamma prior on the scale parameter
`b_lambda`	Hyperparameter of the Gamma prior on on the scale parameter
`R`	Number of MCMC samples after burn-in
`burn_in`	Number of MCMC samples to be discarded as burn-in period
`nstart`	Number of random initializations for the k-means algorithm
`trace`	logical: if true, tracing information on the progress of the algorithm is produced.

Value

`G` A `R x n` matrix including the cluster labels for each MCMC iteration

`lambda` A Rvector of numbers

`loss` A vector of numbers

`G_map` A vector of numbers

`loss_map` A vector of numbers

`Minkowski_gibbs` *K-dissimilarities algorithm with uncertainty quantification*

Description

Perform the Gibbs-sampling for the k-dissimilarities algorithm using the Minkowski distance; see Rigon, T., Herring, A. H. and Dunson, D. B. (2020). This function is complementary to `kdiss`, which is used instead to get a point estimate.

Usage

```
Minkowski_gibbs(
  x,
  k,
  p,
  a_lambda = 0,
  b_lambda = 0,
  R = 1000,
  burn_in = 1000,
  nstart = 10,
  trace = FALSE
)
```

Arguments

- `x` numeric matrix of of the data
- `k` The number of clusters to be considered.
- `p` Power of the Minkowski distance
- `a_lambda` Hyperparameter of the Gamma prior on the scale parameter. The default `a_lambda = 0` leads to an improper prior.
- `b_lambda` Hyperparameter of the Gamma prior on on the scale parameter. The default `a_lambda = 0` leads to an improper prior.
- `R` Number of MCMC samples after burn-in.
- `burn_in` Number of MCMC samples to be discarded as burn-in period.
- `nstart` Number of random initializations for the `kdiss` algorithm, used to initialize the MCMC chain.
- `trace` logical: if true, tracing information on the progress of the algorithm is produced.

Value

`G` Labels of the clusters at each `MCMC` iteration.

`lambda` Numeric vector of the values of `lambda` at each MCMC iteration.

`loss` Numeric vector of the loss function at each MCMC iteration.

`G_map` Labels of the clusters obtained using `kdiss`, representing the maximum a posteriori.

`loss_map` Numeric value of the loss function obtained using `kdiss`, representing the maximized loss.

Index