Package ‘DiscreteGapStatistic’

December 9, 2024

Type Package

Title An Extension of the Gap Statistic for Ordinal/Categorical Data

Version 1.0.0

Description The gap statistic approach is extended to estimate the number of clusters for categorical response format data. This approach and accompanying software is designed to be used with the output of any clustering algorithm and with distances specifically designed for categorical (i.e. multiple choice) or ordinal survey response data.

URL https://github.com/ecortesgomez/DiscreteGapStatistic

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Imports cultevo, magrittr, utils, ggplot2, pheatmap, dplyr,
Polychrome, RColorBrewer, reshape2, tidyr, ComplexHeatmap,
cluster, stats

Suggests kableExtra, knitr, rmarkdown, testthat

Config/testthat/edition 3

Depends R (>= 4.2.0)

NeedsCompilation no

Author Jeffrey Miecznikowski [aut],
Eduardo Cortes [aut, cre] (<https://orcid.org/0000-0002-0966-6488>)

Maintainer Eduardo Cortes <ecortesg@buffalo.edu>

Repository CRAN

Date/Publication 2024-12-09 05:50:02 UTC

Contents

`BhattacharyyaDist` *Bhattacharyya distance core function*

Description

Bhattacharyya distance core function

Usage

```
BhattacharyyaDist(x, adj = 0.01)
```

Arguments

`x`	Matrix
`adj`	Small quantity added to avoid indefinite log(0) values. DEFAULT=0.001

Value

Distance R object

ChisqDist *Chi-square distance core function*

Description

Chi-square distance core function

Usage

```
ChisqDist(x)
```

Arguments

x	Matrix

Value

Distance R object

clusGapDiscr *Discrete application of clusGap Based on the implementation of the function found in the ‘cluster‘ R package.*

Description

Discrete application of clusGap Based on the implementation of the function found in the ‘cluster‘ R package.

Usage

```
clusGapDiscr(
  x,
  clusterFUN,
  K.max,
  B = nrow(x),
  value.range = "DS",
  verbose = interactive(),
  distName = "hamming",
  useLog = TRUE,
  ...
)
```

Arguments

`x` A matrix object specifying category attributes in the columns and observations in the rows.

`clusterFUN` Character string with one of the available clustering implementations. Available options are: ’pam’ (default) from ‘cluster::pam‘, ’diana’ from ‘cluster::diana‘, ’fanny’ from ‘cluster::fanny‘, ’agnes-{average, single, complete, ward, weighted}’ from ‘cluster::fanny‘, ’hclust-{ward.D, ward.D2, single, complete, average, mcquitty, median, centroid}’ from ‘stats::hclust‘, ’kmodes’ from ‘klar::kmodes‘ (‘iter.max = 10‘, ‘weighted = FALSE‘ and ‘fast= TRUE‘). ’kmodes-N’ enables to run the ‘kmodes‘ algorithm with a given number N of iterations where ‘iter.max = N‘.

`K.max` Integer. Maximum number of clusters ‘k‘ to consider

`B` Number of bootstrap samples. By default B = nrow(x).

`value.range` String character vector or a list of character vector with the length matching the number of columns (nQ) of the array. A vector with all categories to consider when bootstrapping the null distribution sample (KS: Known Support option). By DEFAULT vals=NULL, meaning unique range of categories found in the data will be used when drawing the null (DS: Data Support option). If a character vector of categories is provided, these values would be used for the null distribution drawing across the array. If a list with category character vectors is provided, it has to have the same number of columns as the input array. The order of list element corresponds to the array’s columns.

`verbose` Integer or logical. Determines whether progress output should printed while running. By DEFAULT one bit is printed per bootstrap sample.

`distName` String. Name of categorical distance to apply. Available distances: ’bhattacharyya’, ’chisquare’, ’cramerV’, ’hamming’ and ’hellinger’.

`useLog` Logical. Use log function after estimating ‘W.k‘. Following the original formulation ‘useLog=TRUE‘ by default.

`...` optionally further arguments for ‘FUNcluster()‘

Value

a matrix with K.max rows and 4 columns, named "logW", "E.logW", "gap", and "SE.sim", where gap = E.logW - logW, and SE.sim correspond to the standard error of ‘gap‘.

`clusGapDiscr0` *Discrete application of clusGap - core function. Based on the implementation of the function found in the ‘cluster‘ R package.*

Description

Discrete application of clusGap - core function. Based on the implementation of the function found in the ‘cluster‘ R package.

Usage

```
clusGapDiscr0(
  x,
  FUNcluster,
  K.max,
  B = nrow(x),
  value.range = "DS",
  verbose = interactive(),
  distName = "hamming",
  useLog = TRUE,
  Input2Alg = "distMatr",
  ...
)
```

Arguments

`x` A matrix object specifying category attributes in the columns and observations in the rows.

`FUNcluster` a function that accepts as first argument a matrix like ‘x‘; second argument specifies number of ‘k‘ (k=>2) clusters This function should return a list with a component named ‘cluster‘, a vector of length ‘n=nrow(x)‘ of integers from ‘1:k‘ indicating observation cluster assignment. Make sure ‘FUNcluster‘ and ‘Input2Alg‘ agree.

`K.max` Integer. Maximum number of clusters ‘k‘ to consider

`B` Number of bootstrap samples. By default B = nrow(x).

`value.range` String, character vector or a list of character vectors with the length matching the number of columns (nQ) of the array. A vector with all categories to consider when bootstrapping the null distribution sample (KS: Known Support option). By DEFAULT vals=NULL, meaning unique range of categories found in the data will be used when drawing the null (DS: Data Support option). If a character vector of categories is provided, these values would be used for the null distribution drawing across the array. If a list with category character vectors is provided, it has to have the same number of columns as the input array. The order of list element corresponds to the array’s columns.

`verbose` Integer or logical. Determines whether progress output should printed while running. By DEFAULT one bit is printed per bootstrap sample.

`distName` String. Name of categorical distance to apply. Available distances: ’bhattacharyya’, ’chisquare’, ’cramerV’, ’hamming’ and ’hellinger’.

`useLog` Logical. Use log function after estimating ‘W.k‘. Following the original formulation ‘useLog=TRUE‘ by default.

`Input2Alg` Specifies the kind of input provided to the algorithm function in ‘FUNcluster‘. For algorithms that only accept a distance matrix use ‘’distMatr’‘ option (default). For algorithms that require the dataset and a prespecified distance function (e.g. ‘stats::dist‘) use the ‘’distFun’‘ option. This case the distance function is defined internally and determined by parameter ‘distName‘.

`...` optionally further arguments for ‘FUNcluster()‘

Value

a matrix with K.max rows and 4 columns, named "logW", "E.logW", "gap", and "SE.sim", where gap = E.logW - logW, and SE.sim correspond to the standard error of ‘gap‘.

`clusterFunSel` *A function that generates formatted algorithmic functions that can be plugged to enable run a wide variety of clustering algorithm for ‘clusGapDiscr‘ function.*

Description

A function that generates formatted algorithmic functions that can be plugged to enable run a wide variety of clustering algorithm for ‘clusGapDiscr‘ function.

Usage

```
clusterFunSel(clustFun)
```

Arguments

`clustFun`	A character string with the following possible options: ’pam’ (default) from ‘cluster::pam‘, ’diana’ from ‘cluster::diana‘, ’fanny’ from ‘cluster::fanny‘, ’agnes-{average, single, complete, ward, weighted}’ from ‘cluster::agnes‘, ’hclust-{ward.D, ward.D2, single, complete, average, mcquitty, median, centroid}’ from ‘base::hclust‘, ’kmodes’ from ‘klar::kmodes‘ (‘iter.max = 10‘, ‘weighted = FALSE‘ and ‘fast= TRUE‘). ’kmodes-N’ enables to run the ‘kmodes‘ algorithm with a given number N of iterations where ‘iter.max = N‘.

Value

An object of class kmodes as found in ‘klaR‘ packages. An additional component specifies the categorical distance function found in ‘distFun‘.

`concussion` *Concussion Data*

Description

Concussion Data

Usage

```
concussion
```

Format

'data.frame' A data frame with 109 rows and 21 columns. Severity rating recorded as categorical responses from c1 (none) to c7 (severe).

Q1: Headache Headache

Q2: Nausea Nausea

Q3: Balance problems Balance problems

Q4: Dizziness Dizziness

Q5: Fatigue Fatigue

Q6: Sleep more Sleeping more than usual

Q7: Drowsiness Drowsiness

Q8: Sensibility to light Sensibility to light

Q9: Sensibility to noice Sensibility to noice

Q10: Irritability Irritability

Q11: Sadness Sadness

Q12: Nervousness Nervousness/Anxiousness

Q13: More emotional Feeling more emotional

Q14: Feeling slowed down Feeling slowed down

Q15: Feeling mentally foggy Feeling mentally foggy

Q16: Difficulty concentrating Difficulty concentrating

Q17: Difficulty remembering Difficulty remembering

Q18: Visual problem Visual problems

Q19: Confusion Confusion

Q20: Feeling clumsy Feeling clumsy

Q21: Answer slowlier Answer slowlier

`cramersVmod` *Cramer's V modified pairwise vector function based on the function found in lsr package This is simple wrapper of the usual chisq.test fun This is actually an adjusted version of the pi = sqrt(Chisq2/N) guaranteeing that values are within 0 (no association) and 1 (association)*

Description

Cramer's V modified pairwise vector function based on the function found in lsr package This is simple wrapper of the usual chisq.test fun This is actually an adjusted version of the pi = sqrt(Chisq2/N) guaranteeing that values are within 0 (no association) and 1 (association)

Usage

```
cramersVmod(x, y)
```

Arguments

`x`	vector of size n
`y`	vector of size n

Value

numerical value

`CramerV` *Cramer's V core function*

Description

Cramer's V core function

Usage

```
CramerV(X)
```

Arguments

`X`	matrix

Value

Distance matrix

`dissbhattacharyya` *Bhattacharyya's wrapper Function*

Description

Bhattacharyya's wrapper Function

Usage

```
dissbhattacharyya(X, na.rm = TRUE)
```

Arguments

`X`	Matrix
`na.rm`	Remove NAs default=TRUE

Value

Distance R object

disschisquare *Chi-square distance wrapper function*

Description

Chi-square distance wrapper function

Usage

```
disschisquare(X, na.rm = TRUE)
```

Arguments

X	Matrix
na.rm	logical

Value

Distance R object

disscramerv *Cramer's V distance wrapper function*

Description

Cramer's V distance wrapper function

Usage

```
disscramerv(X, na.rm = TRUE)
```

Arguments

X	Matrix
na.rm	logical

Value

Distance R object

`disshamming` *Hamming distance wrapper function Function based on cultevo's package implementation*

Description

Hamming distance wrapper function Function based on cultevo's package implementation

Usage

```
disshamming(X, na.rm = TRUE)
```

Arguments

`X`	matrix
`na.rm`	logical

Value

Distance matrix

`disshellinger` *Hellinger's distance wrapper Function*

Description

Hellinger's distance wrapper Function

Usage

```
disshellinger(X, na.rm = TRUE)
```

Arguments

`X`	Matrix
`na.rm`	logical

Value

Distance R object

`distanceHeat`

sample-to-sample heatmap clustering samples according to a given categorical distance Exploratory tool that helps to visualize/cluster blocks of observations across columns ordered according to given categorical distance. The final output is a clustered distance matrix. This plot is aimed to guide the ‘DiscreteClusGap‘ user to give an idea which type of categorical distance would accommodate better to the inputted data. ‘sample2sampleHeat‘ is based on the ‘pheatmap‘ function from the ‘pheatmap‘ R package. Thus, any parameter found in pheatmap can be specified to ‘sample2sampleHeat‘.

Description

sample-to-sample heatmap clustering samples according to a given categorical distance Exploratory tool that helps to visualize/cluster blocks of observations across columns ordered according to given categorical distance. The final output is a clustered distance matrix. This plot is aimed to guide the ‘DiscreteClusGap‘ user to give an idea which type of categorical distance would accommodate better to the inputted data. ‘sample2sampleHeat‘ is based on the ‘pheatmap‘ function from the ‘pheatmap‘ R package. Thus, any parameter found in pheatmap can be specified to ‘sample2sampleHeat‘.

Usage

```
distanceHeat(
  x,
  distName,
  clustering_method = "complete",
  border_color = NA,
  ...
)
```

Arguments

`x` matrix object or data.frame

`distName` Name of categorical distance to apply.

`clustering_method`
string; clustering method used by pheatmap

`border_color` string; color cell borders. By default, border_color = NA, where no border colors are shown.

`...` other valid arguments in pheatmap function Available distances: ’bhattacharyya’, ’chisquare’, ’cramerV’, ’hamming’ and ’hellinger’.

Value

clustered heatmap

`distancematrix` *Function invoking discrete distance functions*

Description

Function invoking discrete distance functions

Usage

```
distancematrix(X, d, na.rm = TRUE)
```

Arguments

`X`	Matrix where rows are the observations and columns are discrete features
`d`	Name of distance. Distances available: bhattacharyya, chisquare, cramerV, hamming and hellinger
`na.rm`	Remove NAs default=TRUE

Value

R distance object

Examples

```
X = rbind(matrix(paste0("a", rpois(7*5, 1)), nrow=5),
          matrix(paste0("a", rpois(7*5, 3)), nrow=5))
distancematrix(X = X, d = "hellinger")
```

`findK` *Criteria to determine number of clusters k*

Description

Criteria to determine number of clusters k

Usage

```
findK(cG_obj, meth = "Tibs2001SEmax")
```

Arguments

`cG_obj`	Output object obtained from ‘clusGapDiscr‘
`meth`	Method to use to determine optimal k number of clusters.

Value

A numerical value from 1 to K.max, contained in the input ‘cG_obj‘ object.

HellingerDist *Hellinger distance core function*

Description

Hellinger distance core function

Usage

```
HellingerDist(x)
```

Arguments

x	matrix

Value

Distance matrix

kmodesD *Adapted kmodes function to accept any categorical distance based on the function found in ‘klaR:kmodes‘.*

Description

Adapted kmodes function to accept any categorical distance based on the function found in ‘klaR:kmodes‘.

Usage

```
kmodesD(data, modes, distFun, iter.max = 10)
```

Arguments

data	A matrix or data frame of categorical data. Objects have to be in rows, variables in columns.
modes	The number of modes
distFun	Pairwise categorical distance function. A function accepting two categorical vectors.
iter.max	The maximum number of iterations allowed.

Value

An object of class kmodes as found in ‘klaR‘ packages. An additional component specifies the categorical distance function found in ‘distFun‘.

likert.heat.plot2 *Summary Heatmap for categorical/Likert data Heatmap representation summarizing categorical/likert data. Modified version of ‘likert.heat.plot‘ from ‘likert‘ package. Does not allow different categorical ranges across questions. The function outputs a ggplot object where additional layers can be added for customization purposes. The output plot preserves the question order given by columns of ‘x‘.*

Description

Summary Heatmap for categorical/Likert data Heatmap representation summarizing categorical/likert data. Modified version of ‘likert.heat.plot‘ from ‘likert‘ package. Does not allow different categorical ranges across questions. The function outputs a ggplot object where additional layers can be added for customization purposes. The output plot preserves the question order given by columns of ‘x‘.

Usage

```
likert.heat.plot2(
  x,
  allLevels,
  low.color = "white",
  high.color = "blue",
  text.color = "black",
  text.size = 4,
  textLen = 50
)
```

Arguments

`x`	matrix object or data.frame with categorical data. Columns are questions and rows are observations.
`allLevels`	vector with all categorical (ordered) levels.
`low.color`	string; name of color assigned to the first level found in ‘allLevels‘.
`high.color`	string; name of color assigned to the last level found in ‘allLevels‘.
`text.color`	string; text color of numbers within cells.
`text.size`	string; text size for numbers within cells.
`textLen`	string; maximum length of text-length for question labels (column names)

Value

ggplot object.

`mass` *mass data*

Description

mass data

Usage

```
mass
```

Format

‘data.frame‘ Data extracted from the ‘likert‘ R package. Results from an administration of the Math Anxiety Scale Survey. First Column records student gender either Female or Male. All statement answers have 5 possible ordinal categorical items: Strongly Disagree, Disagree, Neutral, Agree, Strongly Agree.

Gender Gender

I find math interesting. Math interesting

I get uptight during math tests. Uptight with math tests

I think that I will use math in the future. Use math in the future

Mind goes blank and I am unable to think clearly when doing my math test. Mind goes blank in math tests

Math relates to my life. Math relates to own life

I worry about my ability to solve math problems. Worry about ability math problem solving

I get a sinking feeling when I try to do math problems. Sinking feeling doing math problems

I find math challenging. Math is challenging

Mathematics makes me feel nervous. Nervousness with math

I would like to take more math classes. Take more math classes

Mathematics makes me feel uneasy. Uneasy feeling with math

Math is one of my favorite subjects. Favorite subject is math

I enjoy learning with mathematics. Enjoy learning math

Mathematics makes me feel confused. Confused with math

Source

<https://rdrr.io/cran/likert/man/mass.html>

ResHeatmap	*Heatmap assuming a given a distance function and a known number of clusters. Function to display a categorical data matrix given a user defined number of clusters 'nCl', a categorical distance 'distName' and a predefined clustering method 'FUNcluster'. The output displays a heatmap separating and color-labelling resulting clusters vertically in the rows and allowing unsupervised clustering on questions in the columns. Each cell is colored according to the categorical values provided or found in the data. The clustergram is based on the 'pheatmap' function from the pheatmap R package. Thus, any parameter found in pheatmap can be specified to 'clusGapDiscrHeat'. This function can be used to examine number of clusters before running 'clusGapDiscrHeat' but also after number of clusters is determined.*

Description

Heatmap assuming a given a distance function and a known number of clusters. Function to display a categorical data matrix given a user defined number of clusters 'nCl', a categorical distance 'distName' and a predefined clustering method 'FUNcluster'. The output displays a heatmap separating and color-labelling resulting clusters vertically in the rows and allowing unsupervised clustering on questions in the columns. Each cell is colored according to the categorical values provided or found in the data. The clustergram is based on the 'pheatmap' function from the pheatmap R package. Thus, any parameter found in pheatmap can be specified to 'clusGapDiscrHeat'. This function can be used to examine number of clusters before running 'clusGapDiscrHeat' but also after number of clusters is determined.

Usage

```
ResHeatmap(
  x,
  nCl,
  distName,
  catVals,
  clusterFUN,
  out = "heatmap",
  seed = NULL,
  clusterNames = NULL,
  prefObs = NULL,
  rowNames = rownames(x),
  filename = NULL,
  outDir = NULL,
  height = 10,
  width = 6
)
```

Arguments

x	matrix object or data.frame

`nCl` number of clusters to plot; if ‘nCl‘ is a permutation vector of the first lN integers will rearrange clusters according to the original given ordering.

`distName` Name of categorical distance to apply. Available distances: ’bhattacharyya’, ’chisquare’, ’cramerV’, ’hamming’ and ’hellinger’.

`catVals` character string vector with (ordered) categorical values

`clusterFUN` Character string with one of the available clustering implementations. Available options are: ’pam’ (default) from ‘cluster::pam‘, ’diana’ from ‘cluster::diana‘, ’fanny’ from ‘cluster::fanny‘. ’agnes-{average, single, complete, ward, weighted}’ from ‘cluster::agnes‘, ’hclust-{ward.D, ward.D2, single, complete, average, mcquitty, median, centroid}’ from ‘stats::hclust‘, ’kmodes’ from ‘klar::kmodes‘ (‘weighted = FALSE‘ and ‘fast= TRUE‘).

`out` Specifies the desired output between "heatmap" (default; produce a heatmap), "clusters" (return a ‘data.frame‘ with clustering assignments) or "clustersReord" (return a ‘data.frame‘ with reorganized clusters)

`seed` Seed number.

`clusterNames` Either ‘null‘ or ’renumber’. When ‘nCl‘ is a numerical vector, the cluster ordering is rearranged. ‘NULL‘ leaves cluster names as their original cluster assignment. ’renumber’ respects the rearrangements but relabels the cluster numbers from top to bottom in ascending order.

`prefObs` character string vector of length 1 with a prefix for the observations, in case they come unlabelled or the user wants to anomymize sample IDs.

`rowNames` character vector with names of rows according to ‘x‘. By default, ‘rownames(x)‘ will be printed in the plot. ‘rowNames=NULL‘ prevents from showing names. ‘prefObs‘ option takes precedence if is different to ‘NULL‘.

`filename` character string with name of file output

`outDir` character string with the directory path to save output file

`height` numeric height of output plot in inches

`width` numeric width of output plot in inches

Value

png file or ComplexHeatmap object

SimData *Simulate Data*

Description

Simulate Data

Usage

```
SimData(N, nQ, pi)
```

Arguments

N	Integer. Number of observations.
nQ	Integer. Number of questions.
pi	Numeric vector. Vector of probabilities adding up to 1; it is recommended that names of elements are character strings. Alternatively, pi can be list of vectors as previously described with length equal to ‘nQ‘. Notice that the list elements need not have same vector names. The order of pi vectors in the list will be reflected in the resulting simulated matrix. This alternative ideally assumes that questions are independently distributed.

Value

N x nQ matrix with simulated categories distributed according to vector pi

Examples

```
Pix <- setNames(c(0.1, 0.2, 0.3, 0.4, 0), paste0('a', 1:5))
X <- SimData(N=10, nQ=5, Pix)
head(X)

Piy <- setNames(c(0.3, 0.2, 0.4, 0, 0.1), paste0('a', 1:5))
Y <- SimData(N=10, nQ=3, Piy)
head(Y)

PiZ <- list(x1 = Pix, x2 = Pix, y1 = Piy, y2 = Piy)
Z <- SimData(N=10, nQ=length(PiZ), PiZ)
```

Index