Package 'muHVT'

January 21, 2020

Type Package

Date 2020-01-09

Title Constructing Hierarchical Voronoi Tessellations and Overlay Heatmap for Data Analysis

Version 1.0.1

Description Constructing hierarchical voronoi tessellations for a given data set and overlay heatmap for variables at various levels of the tessellations for in-depth data analysis. See <https://en.wikipedia.org/wiki/Voronoi_diagram> for more information. Credits to Mu Sigma for their continuous support throughout the development of the package.

License Apache License 2.0

Encoding UTF-8

Imports MASS, deldir, grDevices, splancs, sp, conf.design, Hmisc, dplyr, purrr, gtools, magrittr, plyr, polyclip, rgeos, ggplot2

Depends R (>= 3.5.0)

BugReports https://github.com/Mu-Sigma/muHVT/issues

URL https://github.com/Mu-Sigma/muHVT

LazyData true

RoxygenNote 7.0.2

Suggests knitr, rmarkdown, testthat,geozoo,plotly,kableExtra

VignetteBuilder knitr

NeedsCompilation no

Author Sangeet Moy Das [aut],
Zubin Dowlaty [aut],
Avinash Joshi [aut],
Meet Dave [ctb],
Shubhra Prakash [ctb],
Mu Sigma, Inc. [cre]

Maintainer ``Mu Sigma, Inc." <ird.experiencelab@mu-sigma.com>

Repository CRAN

Date/Publication 2020-01-21 10:30:02 UTC

R topics documented:

`getOptimalCentroids` *getOptimalCentroids*

Description

Get Optimal Centroids

Usage

```
getOptimalCentroids(
  x,
  iter.max,
  algorithm,
  nclust,
  distance_metric,
  error_metric,
  quant.err
)
```

Arguments

`x`	Data Frame. A dataframe of multivariate data. Each row corresponds to an observation, and each column corresponds to a variable. Missing values are not accepted.
`iter.max`	The max number of iterations the the getOptimalCentroid function will run to get the optimal number of centroids
`algorithm`	String. The type of algorithm used for quantization. Available algorithms are Hartigan and Wong, "Lloyd", "Forgy", "MacQueen". (default is "Hartigan-Wong")
`nclust`	Numeric. Indicating the number of nodes per hierarchy.
`distance_metric`	character. The distance metric can be 'L1_Norm" or "L2_Norm". L1_Norm is selected by default.
`error_metric`	character. The error metric can be "mean" or "max". mean is selected by default
`quant.err`	Numeric. The quantization error for the algorithm.

Details

The raw data is first scaled and this scaled data is supplied as input to the vector quantization algorithm. Vector quantization technique uses a parameter called quantization error. This parameter acts as a threshold and determines the number of levels in the hierarchy. It means that, if there are 'n' number of levels in the hierarchy, then all the clusters formed till this level will have quantization error equal or greater than the threshold quantization error. The user can define the number of clusters in the first level of hierarchy and then each cluster in first level is sub-divided into the same number of clusters as there are in the first level. This process continues and each group is divided into smaller clusters as long as the threshold quantization error is met. The output of this technique will be hierarchically arranged vector quantized data.

Value

`clusters`	List. A list showing each ID assigned to a cluster.
`nodes.clust`	List. A list corresponding to nodes' details.
`idnodes`	List. A list of ID and segments similar to `nodes.clust` with additional columns for nodes ID.
`error.quant`	List. A list of quantization error for all levels and nodes.
`plt.clust`	List. A list of logical values indicating if the quantization error was met.
`summary`	Summary. Output table with summary.

Author(s)

Sangeet Moy Das <sangeet.das@mu-sigma.com>

`hvq` *hvq*

Description

Hierarchical Vector Quantization

Usage

```
hvq(
  x,
  nclust = 15,
  depth = 3,
  quant.err = 0.2,
  algorithm = c("Hartigan-Wong", "Lloyd", "Forgy", "MacQueen"),
  distance_metric = c("L1_Norm", "L2_Norm"),
  error_metric = c("mean", "max")
)
```

Arguments

x	Data Frame. A dataframe of multivariate data. Each row corresponds to an observation, and each column corresponds to a variable. Missing values are not accepted.
nclust	Numeric. Indicating the number of nodes per hierarchy.
depth	Numeric. Indicating the hierarchy depth (or) the depth of the tree (1 = no hierarchy, 2 = 2 levels, etc..)
quant.err	Numeric. The quantization error for the algorithm.
algorithm	String. The type of algorithm used for quantization. Available algorithms are Hartigan and Wong, "Lloyd", "Forgy", "MacQueen". (default is "Hartigan-Wong")
distance_metric	character. The distance metric can be 'L1_Norm" or "L2_Norm". L1_Norm is selected by default.
error_metric	character. The error metric can be "mean" or "max". mean is selected by default

Details

The raw data is first scaled and this scaled data is supplied as input to the vector quantization algorithm. Vector quantization technique uses a parameter called quantization error. This parameter acts as a threshold and determines the number of levels in the hierarchy. It means that, if there are 'n' number of levels in the hierarchy, then all the clusters formed till this level will have quantization error equal or greater than the threshold quantization error. The user can define the number of clusters in the first level of hierarchy and then each cluster in first level is sub-divided into the same number of clusters as there are in the first level. This process continues and each group is divided into smaller clusters as long as the threshold quantization error is met. The output of this technique will be hierarchically arranged vector quantized data.

Value

clusters	List. A list showing each ID assigned to a cluster.
nodes.clust	List. A list corresponding to nodes' details.
idnodes	List. A list of ID and segments similar to nodes.clust with additional columns for nodes ID.
error.quant	List. A list of quantization error for all levels and nodes.
plt.clust	List. A list of logical values indicating if the quantization error was met.
summary	Summary. Output table with summary.

Author(s)

Sangeet Moy Das <Sangeet.Das@mu-sigma.com>

See Also

hvtHmap

Examples

```
data("USArrests",package="datasets")
hvqOutput = hvq(USArrests, nclust = 3, depth = 3, quant.err = 0.2,
distance_metric = 'L1_Norm', error_metric = 'mean')
```

HVT *Constructing Hierarchical Voronoi Tessellations*

Description

Main function to construct hierarchical voronoi tessellations.

Usage

```
HVT(
  dataset,
  nclust = 15,
  depth = 3,
  quant.err = 0.2,
  projection.scale = 10,
  normalize = TRUE,
  distance_metric = c("L1_Norm", "L2_Norm"),
  error_metric = c("mean", "max")
)
```

Arguments

dataset	Data frame. A data frame with different columns is given as input.
nclust	Numeric. An integer indicating the number of clusters per hierarchy (level)
depth	Numeric. An integer indicating the number of levels. (1 = No hierarchy, 2 = 2 levels, etc ...)
quant.err	Numeric. A number indicating the quantization error treshold.
projection.scale	Numeric. A number indicating the scale factor for the tesselations so as to visualize the sub-tesselations well enough.
normalize	Logical. A logical value indicating if the columns in your dataset should be normalized. Default value is TRUE.
distance_metric	character. The distance metric can be 'Euclidean" or "Manhattan". Euclidean is selected by default.
error_metric	character. The error metric can be "mean" or "max". mean is selected by default

Details

This is the main function to construct hierarchical voronoi tessellations. The hvq function is called from this function. The output of the hvq function is hierarchical clustered data which will be the input for constructing tessellations. The data is then represented in 2d coordinates and the tessellations are plotted using these coordinates as centroids. For subsequent levels, transformation is performed on the 2d coordinates to get all the points within its parent tile. Tessellations are plotted using these transformed points as centroids. The lines in the tessellations are chopped in places so that they do not protrude outside the parent polygon. This is done for all the subsequent levels.

Value

A list that contains the hierarchical tesselation information. This list has to be given as input argument to plot the tessellations.

[[1]]	List. Information about the tesselation co-ordinates - level wise
[[2]]	List. Information about the polygon co-ordinates - level wise
[[3]]	List. Information about the hierarchical vector quantized data - level wise

Author(s)

Sangeet Moy Das <sangeet.das@mu-sigma.com>

See Also

plotHVT
hvtHmap

Examples

```
data(USArrests)
hvt.results <- list()
hvt.results <- HVT(USArrests, nclust = 15, depth = 1, quant.err = 0.2,
                   distance_metric = "L1_Norm", error_metric = "mean",
                   projection.scale = 10, normalize = TRUE)
plotHVT(hvt.results, line.width = c(0.8), color.vec = c('#141B41'),
        maxDepth = 1)

hvt.results <- list()
hvt.results <- HVT(USArrests, nclust = 3, depth = 3, quant.err = 0.2,
                   distance_metric = "L1_Norm", error_metric = "mean",
                   projection.scale = 10, normalize = TRUE)
plotHVT(hvt.results, line.width = c(1.2,0.8,0.4), color.vec = c('#141B41','#0582CA','#8BA0B4'),
        maxDepth = 3)
```

hvtHmap *Heat Map over Hierarchical Voronoi Tessellations*

Description

Main function to construct heatmap overlay for hierarchical voronoi tessellations.

Usage

```
hvtHmap(
  hvt.results,
  dataset,
  child.level,
  hmap.cols,
  color.vec = NULL,
  line.width = NULL,
  centroid.size = 3,
  pch = 21,
  palette.color = 6,
  previous_level_heatmap = TRUE,
  show.points = FALSE,
  asp = 1,
  ask = TRUE,
  tess.label = NULL,
  quant.error.hmap = NULL,
  nclust.hmap = NULL,
  label.size = 0.5,
  ...
)
```

Arguments

hvt.results	List. A list of hvt.results obtained from the HVT function.
dataset	Data frame. The input data set.
child.level	Numeric. Indicating the level for which the heat map is to be plotted.
hmap.cols	Numeric or Character. The column number of column name from the dataset indicating the variables for which the heat map is to be plotted.
color.vec	Vector. A color vector such that length(color.vec) = (child.level - 1). (default = NULL)
line.width	Vector. A line width vector such that length(line.width) = (child.level - 1). (default = NULL)
centroid.size	Numeric. Indicating the centroid size of the first level. (default = 3)
pch	Numeric. Indicating the centroid's symbol type. (default = 21)
palette.color	Numeric. Indicating the heat map color palette. 1 - rainbow, 2 - heat.colors, 3 - terrain.colors, 4 - topo.colors, 5 - cm.colors, 6 - seas color. (default = 6)

previous_level_heatmap
: Logical. If TRUE, the heatmap of previous level will be overlayed on the heatmap of selected level. If #' FALSE, the heatmap of only selected level will be plotted

show.points
: Logical. Indicating if the centroids should be plotted on the tesselations. (default = FALSE)

asp
: Numeric. Indicating the aspect ratio type. For flexible aspect ratio set, asp = NA. (default = 1)

ask
: Logical. If TRUE (and the R session is interactive) the user is asked for input, before a new figure is drawn. (default = TRUE)

tess.label
: Vector. A vector for labelling the tessellations. (default = NULL)

quant.error.hmap
: Numeric. A number indicating the quantization error treshold.

nclust.hmap
: Numeric. An integer indicating the number of clusters per hierarchy (level)

label.size
: Numeric. The size by which the tessellation labels should be scaled. (default = 0.5)

...
: The ellipsis is passed to it as additional argument. (Used internally)

Details

The output of the HVT function has all the required information about the HVT. Now a heat map is overlayed over this HVT. The user defines the level and also those variables of the data for which the heat map is to be plotted. Now for each variable a separate heat map is plotted. The plot area is divided into 2 screens where the first screen is relatively large and will have the heat map. The second screen is small and contains the gradient scale. To plot the heat map, the data is first normalized. The gradient scale is divided into 'n' regions(500 is the set default). Using the normalized data, the different regions into which the data items fall are found. Each data item is now having a region on the gradient scale. This color is filled in the tile corresponding to the data item. This procedure is done for all the tiles for that level to get the complete heat map. Once the heat map is ready, the higher level tessellations are plotted to represent the hierarchies. The size of the centroids, the thickness of the lines and the color of the tessellation lines can be given as input by the user. Appropriate values for these parameters should be given to identify the hierarchies properly. In the second screen the gradient scale is plotted. The heat maps and hierarchical tessellations are obtained for all the desired variables.

Author(s)

Sangeet Moy Das <sangeet.das@mu-sigma.com>

See Also

plotHVT

Examples

```
data(USArrests)
hvt.results <- list()
```

```
hvt.results <- HVT(USArrests, nclust = 6, depth = 1, quant.err = 0.2,
                   distance_metric = "L1_Norm", error_metric = "mean",
                   projection.scale = 10, normalize = TRUE)
hvtHmap(hvt.results, USArrests, child.level = 1,hmap.cols = 'Murder',
        line.width = c(0.2), color.vec = c('#141B41'),palette.color = 6,
        quant.error.hmap = 0.2,nclust.hmap = 6)

hvt.results <- list()
hvt.results <- HVT(USArrests, nclust = 3, depth = 3, quant.err = 0.2,
                   distance_metric = "L1_Norm", error_metric = "mean",
                   projection.scale = 10, normalize = TRUE)
hvtHmap(hvt.results, USArrests, child.level = 3,hmap.cols = 'Quant.Error',
        line.width = c(1.2,0.8,0.4),color.vec = c('#141B41','#0582CA','#8BA0B4'),
        palette.color = 6,quant.error.hmap = 0.2,nclust.hmap = 3)
```

---

plotHVT *Plot the hierarchical tesselations.*

Description

Main plotting function to construct hierarchical voronoi tessellations.

Usage

```
plotHVT(
  hvt.results,
  line.width,
  color.vec,
  pch1 = 21,
  centroid.size = 3,
  title = NULL,
  maxDepth = NULL
)
```

Arguments

hvt.results	List. A list containing the ouput of HVT function which has the details of the tessellations to be plotted.
line.width	Numeric Vector. A vector indicating the line widths of the tessellation boundaries for each level.
color.vec	Vector. A vector indicating the colors of the boundaries of the tessellations at each level.
pch1	Numeric. Symbol type of the centroids of the tessellations (parent levels). Refer points. (default = 21)
centroid.size	Numeric. Size of centroids of first level tessellations. (default = 3)
title	String. Set a title for the plot. (default = NULL)
maxDepth	Numeric. An integer indicating the number of levels. (default = NULL)

Author(s)

Sangeet Moy Das <sangeet.das@mu-sigma.com>

See Also

HVT
hvtHmap

Examples

```
data("USArrests",package="datasets")

hvt.results <- list()
hvt.results <- HVT(USArrests, nclust = 3, depth = 3, quant.err = 0.2,
                   distance_metric = "L1_Norm", error_metric = "mean",
                   projection.scale = 10, normalize = TRUE)
plotHVT(hvt.results, line.width = c(1.2,0.8,0.4), color.vec = c('#141B41','#0582CA','#8BA0B4'),
        maxDepth = 3)
```

---

predictHVT *Predict which cell and what level each point in the test dataset belongs to*

Description

Main function to predict cell path of new datapoints

Usage

```
predictHVT(
  data,
  hvt.results,
  hmap.cols = NULL,
  child.level = 1,
  quant.error.hmap = NULL,
  nclust.hmap = NULL,
  line.width = NULL,
  color.vec = NULL,
  ...
)
```

Arguments

data	List. A dataframe containing test dataset. The dataframe should have atleast one variable used while training. The variables from this dataset can also be used to overlay as heatmap

hvt.results	A list of hvt.results obtained from HVT function while performing hierarchical vector quantization on train data
hmap.cols	- The column number of column name from the dataset indicating the variables for which the heat map is to be plotted.(Default = #' NULL). A heatmap won't be plotted if NULL is passed
child.level	A number indicating the level for which the heat map is to be plotted.(Only used if hmap.cols is not NULL)
quant.error.hmap	Numeric. A number indicating the quantization error threshold.
nclust.hmap	Numeric. An integer indicating the number of clusters per hierarchy
line.width	Vector. A line width vector such that length(line.width) = (child.level - 1). (default = NULL)
color.vec	Vector. A color vector such that length(color.vec) = (child.level - 1). (default = NULL)
...	color.vec and line.width can be passed from here

Author(s)

Sangeet Moy Das <sangeet.das@mu-sigma.com>

See Also

HVT
hvtHmap

Examples

```
data(USArrests)
#Split in train and test

train <- USArrests[1:40,]
test <- USArrests[41:50,]

hvt.results <- list()
hvt.results <- HVT(train, nclust = 3, depth = 2, quant.err = 0.2,
                   distance_metric = "L1_Norm", error_metric = "mean",
                   projection.scale = 10, normalize = TRUE)

predictions <- predictHVT(test,hvt.results,hmap.cols = "Quant.Error", child.level=2,
                      quant.error.hmap = 0.2,nclust.hmap = 3,line.width = c(1.2,0.8,0.4),
                          color.vec = c('#141B41','#0582CA','#8BA0B4'))
print(predictions$predictions)
```

sammonsProjection *sammonsProjection*

Description

This is a wrapper for the sammon function of the MASS package for non-metric multidimensional scaling

Usage

```
sammonsProjection(
  d,
  y = stats::cmdscale(d, k),
  k = 2,
  niter = 100,
  trace = TRUE,
  magic = 0.2,
  tol = 1e-04
)
```

Arguments

d	distance structure of the form returned by dist, or a full, symmetric matrix. Data are assumed to be dissimilarities or relative distances, but must be positive except for self-distance. This can contain missing values.
y	An initial configuration. If none is supplied, cmdscale is used to provide the classical solution. (If there are missing values in d, an initial configuration must be provided.) This must not have duplicates.
k	The dimension of the configuration.
niter	The maximum number of iterations.
trace	Logical for tracing optimization. Default TRUE.
magic	initial value of the step size constant in diagonal Newton method.
tol	Tolerance for stopping, in units of stress.

Details

This chooses a two-dimensional configuration to minimize the stress, the sum of squared differences between the input distances and those of the configuration, weighted by the distances, the whole sum being divided by the sum of input distances to make the stress scale-free.

An iterative algorithm is used, which will usually converge in around 50 iterations. As this is necessarily an O(n^2) calculation, it is slow for large datasets. Further, since the configuration is only determined up to rotations and reflections (by convention the centroid is at the origin), the result can vary considerably from machine to machine. In this release the algorithm has been modified by adding a step-length search (magic) to ensure that it always goes downhill.

Value

points	A two-column vector of the fitted configuration.
stress	The final stress achieved.

Examples

```
require(MASS)
swiss.x <- as.matrix(swiss[, -1])
swiss.sam <- sammonsProjection(dist(swiss.x))
```

VQ_codebookSplit *VQ_codebookSplit*

Description

Vector Quantization by codebook split method

Usage

```
VQ_codebookSplit(dataset, quant.err = 0.5, epsilon = NULL)
```

Arguments

dataset	Matrix. A matrix of multivariate data. Each row corresponds to an observation, and each column corresponds to a variable. Missing values are not accepted.
quant.err	Numeric. The quantization error for the algorithm.
epsilon	Numeric. The value to offset the codebooks during the codebook split. Default is NULL, in which case the value is set to quant.err parameter.

Details

Performs Vector Quantization by codebook split method. Initially, the entire dataset is considered to be one cluster where the codebook is the mean of the cluster. The quantization criteria is checked and the codebook is split such that the new codebooks are (codebook+epsilon) and (codebook-epsilon). The observations are reassigned to these new codebooks based on the nearest neighbour condition and the means recomputed for the new clusters. This is done iteratively until all the clusters meet the quantization criteria.

Value

`clusters`	List. A list showing each ID assigned to a cluster.
`nodes.clust`	List. A list corresponding to nodes' details.
`idnodes`	List. A list of ID and segments similar to `nodes.clust` with additional columns for nodes ID.
`error.quant`	List. A list of quantization error for all levels and nodes.
`plt.clust`	List. A list of logical values indicating if the quantization error was met.
`summary`	Summary. Output table with summary.

Author(s)

Sangeet Moy Das <sangeet.das@mu-sigma.com>

See Also

`hvtHmap`

Examples

```
data("iris",package="datasets")
iris <- iris[,1:2]

vqOutput = VQ_codebookSplit(iris, quant.err = 0.5)
```

Index