Package: GridOnClusters (via r-universe)

October 8, 2024

Type Package

Title Cluster-Preserving Multivariate Joint Grid Discretization

Version 0.1.0.1

Date 2024-05-10

Author Jiandong Wang [aut], Sajal Kumar [aut] (<https://orcid.org/0000-0003-0930-1582>), Joe Song [aut, cre] (<https://orcid.org/0000-0002-6883-6547>)

Maintainer Joe Song <joemsong@cs.nmsu.edu>

Description Discretize multivariate continuous data using a grid that captures the joint distribution via preserving clusters in the original data (Wang et al. 2020) <doi:10.1145/3388440.3412415>. Joint grid discretization is applicable as a data transformation step to prepare data for model-free inference of association, function, or causality.

Imports Rcpp, Ckmeans.1d.dp, cluster, fossil, dqrng, mclust, Rdpack, plotrix

Suggests FunChisq, knitr, testthat (>= 3.0.0), rmarkdown

RdMacros Rdpack

License LGPL (>= 3)

Encoding UTF-8

LinkingTo Rcpp

RoxygenNote 7.3.1

NeedsCompilation yes

Config/testthat/edition 3

VignetteBuilder knitr

Date/Publication 2024-05-10 16:13:11 UTC

Repository https://joemsong.r-universe.dev

RemoteUrl https://github.com/cran/GridOnClusters

RemoteRef HEAD

RemoteSha e8e23f4150fe214a7529bead7f502f00dbf8768d

Contents

`cluster` *Cluster Multivariate Data*

Description

The function obtains clusters from data using the given number of clusters, which may be a range.

Usage

```
cluster(data, k, method)
```

Arguments

`data`	input continuous multivariate data
`k`	the number(s) of clusters
`method`	the method for clustering

`discretize.jointly` *Discretize Multivariate Continuous Data by a Cluster-Preserving Grid*

Description

Discretize multivariate continuous data using a grid that captures the joint distribution via preserving clusters in the original data

Usage

```
discretize.jointly(
  data,
  k = c(2:10),
  min_level = 1,
  cluster_method = c("Ball+BIC", "kmeans+silhouette", "PAM"),
  grid_method = c("Sort+split", "MultiChannel.WUC"),
  cluster_label = NULL
)
```

Arguments

data	a matrix containing two or more continuous variables. Columns are variables, rows are observations.
k	either an integer, a vector of integers, or Inf, specifying different ways to find clusters in data. The default is a vector containing integers from 2 to 10. If 'k' is a single number, data will be grouped into into exactly 'k' clusters. If 'k' is an integer vector, an optimal 'k' is chosen from among the integers, that maximizes the average silhouette width. If 'k' is set to Inf, an optimal k is chosen among 2 to nrow(data). If cluster_label is specified, k is ignored.
min_level	integer or vector, signifying the minimum number of levels along each dimension. If a vector of size ncol(data), then each element will be mapped 1:1 to each dimension in order. If an integer, then all dimensions will have the same minimum number of levels.
cluster_method	the clustering method to be used. Ignored if cluster labels are given "kmeans+silhouette" will use k-means to cluster data and the average Silhouette score to select the number of clusters k. "Ball+BIC" will use Mclust (modelNames = "VII") to cluster data and BIC score to select the number of cluster k.
grid_method	the discretization method to be used. "Sort+split" will sort the cluster by cluster mean in each dimension and then split consecutive pairs only if the sum of the error rate of each cluster is less than or equal to 50 in a certain dimension. The maximum number of lines is the number of clusters minus one. "MultiChannel.WUC" will split each dimension by weighted with-in cluster sum of squared distances by "Ckmeans.1d.dp::MultiChannel.WUC". Applied in each projection on each dimension. The channel of each point is defined by its multivariate cluster label.
cluster_label	a vector of user-specified cluster labels for each observation in data. The user is free to choose any clustering. If unspecified, k-means clustering is used by default.

Details

The function implements algorithms described in (Wang et al. 2020).

Value

A list that contains four items:

D	a matrix that contains the discretized version of the original data. Discretized values are one(1)-based.
grid	a list of vectors containing decision boundaries for each variable/dimension.
clabels	a vector containing cluster labels for each observation in data.
csimilarity	a similarity score between clusters from joint discretization D and cluster labels clabels. The score is the adjusted Rand index.

Author(s)

Jiandong Wang, Sajal Kumar and Mingzhou Song

References

Wang J, Kumar S, Song M (2020). "Joint Grid Discretization for Biological Pattern Discovery." In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. ISBN 9781450379649, doi:10.1145/3388440.3412415.

See Also

See Ckmeans.1d.dp for discretizing univariate continuous data.

Examples

```
# using a specified k
x = rnorm(100)
y = sin(x)
z = cos(x)
data = cbind(x, y, z)
discretized_data = discretize.jointly(data, k=5)$D

# using a range of k
x = rnorm(100)
y = log1p(abs(x))
z = tan(x)
data = cbind(x, y, z)
discretized_data = discretize.jointly(data, k=c(3:10))$D

# using k = Inf
x = c()
y = c()
mns = seq(0,1200,100)
for(i in 1:12){
  x = c(x,runif(n=20, min=mns[i], max=mns[i]+20))
  y = c(y,runif(n=20, min=mns[i], max=mns[i]+20))
}
data = cbind(x, y)
discretized_data = discretize.jointly(data, k=Inf)$D

# using an alternate clustering method to k-means
library(cluster)
x = rnorm(100)
y = log1p(abs(x))
z = sin(x)
data = cbind(x, y, z)

# pre-cluster the data using partition around medoids (PAM)
cluster_label = pam(x=data, diss = FALSE, metric = "euclidean", k = 5)$clustering
discretized_data = discretize.jointly(data, cluster_label = cluster_label)$D
```

plot.GridOnClusters *Plotting the continuous data along with cluster-preserving Grid*

Description

Plots examples of jointly discretizing continuous data based on grids that preserve clusters in the original data.

Usage

```
## S3 method for class 'GridOnClusters'
plot(
  x,
  xlab = NULL,
  ylab = NULL,
  main = NULL,
  main.table = NULL,
  sub = NULL,
  pch = 19,
  ...
)
```

Arguments

x	the result generated by discretize.jointly
xlab	the horizontal axis label
ylab	the vertical axis label
main	the title of the clustering scatter plots
main.table	the title of the discretized data plots
sub	the subtitle
pch	the symbol for points on the scatter plots
...	additional graphical parameters

plotGOCpatterns *(OBOSOLETE) Plotting the continuous data along with cluster-preserving Grid*

Description

Plots examples of jointly discretizing continuous data based on grids that preserve clusters in the original data.

Usage

```
plotGOCpatterns(data, res)
```

Arguments

data	the input continuous data matrix
res	the result generated by discretize.jointly

Index