

Package ‘GridOnClusters’

July 18, 2026

Type Package

Title Cluster-Preserving Multivariate Joint Grid Discretization

Version 0.4.1

Date 2026-07-18

Depends R (>= 3.5.0)

Author Jiandong Wang [aut] (ORCID: <<https://orcid.org/0009-0006-2040-7870>>),
Sajal Kumar [aut] (ORCID: <<https://orcid.org/0000-0003-0930-1582>>),
Joe Song [aut, cre] (ORCID: <<https://orcid.org/0000-0002-6883-6547>>)

Maintainer Joe Song <joemsong@nmsu.edu>

Description Discretize multivariate continuous data using a grid to capture the joint distribution that preserves clusters in original data (Wang et al 2020) <[doi:10.1145/3388440.3412415](https://doi.org/10.1145/3388440.3412415)>. It can handle both labeled or unlabeled data. Joint grid discretization can 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 (>= 2.1.0), rmarkdown

RdMacros Rdpack

License LGPL (>= 3)

Encoding UTF-8

LinkingTo BH, Rcpp, RcppArmadillo

NeedsCompilation yes

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

Repository CRAN

Date/Publication 2026-07-18 12:40:02 UTC

Contents

ballclust	2
cluster	3
discretize.jointly	3
gen_simdata	7
hashed_epsilon	8
plot.GridOnClusters	8
plot.GOCpatterns	9
Index	11

ballclust	<i>Fast VII Gaussian Mixture Model</i>
-----------	--

Description

Fast VII Gaussian Mixture Model

Usage

```
ballclust(  
  data,  
  G = NULL,  
  threads = parallel::detectCores() - 1,  
  min_pts = 5,  
  verbose = FALSE  
)
```

Arguments

data	A numeric matrix or data frame.
G	An integer vector specifying the numbers of mixture components. If NULL, defaults to 1:9.
threads	Number of threads for OpenMP parallelization.
min_pts	an integer to specify the minimum number of points inside a cluster. Default: 5
verbose	logical. if TRUE, print out additional messages. Default: FALSE

Value

An object of class ballclust, mimicking mclust.

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, noise, min_pts, nthread)
```

Arguments

data	input continuous multivariate data
k	the number(s) of clusters
method	the method for clustering
noise	adding jitter noise to the data or not
min_pts	an integer to specify the minimum number of points inside a cluster.
nthread	an integer to specify the number of threads

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,
  max_level = 100,
  cluster_method = c("Ball+BIC", "mclust_VII", "kmeans+silhouette", "PAM"),
  grid_method = c("DP approx likelihood 1-way", "DP approx likelihood 2-way",
    "DP exact likelihood", "DP Compressed majority", "DP", "Sort+split",
    "MultiChannel.WUC"),
  eval_method = c("ARI", "purity", "upslion", "CAIR"),
  cluster_label = NULL,
  cutoff = 0,
  entropy = FALSE,
```

```

noise = FALSE,
dim_reduction = FALSE,
scale = FALSE,
variance = 0.5,
min_pts = 5,
nthread = 1,
debug = FALSE
)

```

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 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.
max_level	integer or vector, signifying the maximum number of levels along each dimension. Works in the same way as min_level. Max_level will be set to the smaller one between the number of compressed zones and itself, if the grid_method is set to "dp compressed likelihood" or "dp compressed majority"
cluster_method	the clustering method to be used. Ignored if cluster labels are given. "kmmeans+silhouette" will use k-means to cluster data and the average Silhouette score to select the number of clusters k. "Ball+BIC": Uses a custom, multithreaded spherical clustering method. It builds isotropic, variable-radius clusters and uses the BIC score to automatically select the optimal k. "mclust_VII": Uses the mclust package (modelName = "VII") to perform spherical clustering, also using BIC for k selection. "PAM" pam is partition around medoids See Details for more information.
grid_method	a character string to specify the discretization method. Default: "DP approx likelihood 1-way". The methods can be roughly separate into three different categories: by cluster likelihood, by density, and by SSE (Sum of Squared Errors). See Details for more information.
eval_method	a character string to specify the method to evaluate quality of discretized data
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.

cutoff	a floating number. Tell the quality to decide whether cut that line or not. If the statistic, a value to evaluate the quality of a cut, is smaller than cut_off, no cut will be performed. Only will be used when the grid_method is set to "DP" or "DP Compressed majority"
entropy	Boolean variable decide whether using entropy or likelihood, default value is false.
noise	adding jitter noise to the original data, only will be used when the cluster_method set to "Ball+BIC". When the data are very dense, adding noise could improve the Mclust's result.
dim_reduction	a logical to turn on/off dimension reduction. Default: FALSE.
scale	a logical to specify linear scaling of the variable in each dimension. Default: FALSE.
variance	a numeric value to specify noise variance to be added to the data
min_pts	an integer to specify the minimum number of points inside a cluster. Default: 5
nthread	Integer. Number of CPU threads to use. Automatically adjusted if invalid or exceeds available cores. Default: 1
debug	logical. If TRUE, it enables the debug mode. Default: FALSE

Details

The function implements both published algorithms described in (Wang et al. 2020) and new algorithms for multivariate discretization.

The grid methods provided is roughly in three different kinds.

- By Density
 - "Sort+split" (Wang et al. 2020) 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%. It is possible that no grid line will be added in a certain dimension. The maximum number of lines is the number of clusters minus one.
- By SSE (Sum of Squared Errors)
 - "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.
 - "DP" will order the label by the data in each dimension and then cut the data into max of max_level bins, then evaluating the quality of each cut to find the best number of bins.
 - "DP Compressed majority" will order the label by the data in each dimension and then compress the labels that neighbored by the same label to avoid trying discretization within consecutive points of the same cluster label to greatly reduce runtime of dynamic programming. Then cut the data into max of max_level bins, and evaluating the quality of each cut by the majority of the data has been spliced clearly to find the best number of bins.
- By cluster likelihood
 - "DP exact likelihood" will order the label by the data in each dimension and then compress the labels that neighbored by the same label to avoid trying discretization within consecutive points of the same cluster label to greatly reduce runtime of dynamic programming. Then cut the data into max of max_level bins.

- "DP approx likelihood 1-way" is a speed up version of the "DP exact likelihood" method, but no optimal will be guaranteed.
- "DP approx likelihood 2-way" is a bidirectional variant of the "DP approx likelihood" method. It performs approximate dynamic programming cuts in both the forward and backward directions and selects the better of the two results. This approach provides additional robustness compared to the one-directional version, but optimality is still not guaranteed.

The clsuter methods provided:

- `"Ball+BIC"`: Employs a custom, high-performance spherical clustering

algorithm. It models clusters as isotropic spheres (equal shape, variable volume) and automatically selects the optimal number of clusters (k) by maximizing the Bayesian Information Criterion (BIC). This implementation supports multithreading for scalable execution on high-dimensional data.

- `"mclust_VII"`: Utilizes the `mclust` package to fit a Gaussian

Mixture Model with a spherical, unequal volume covariance structure (modelName = "VII"). It relies on the BIC score to automatically determine the optimal number of components.

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.

Note

The default value of `grid_method` is changed from "Sort+Split" (up to released package version 0.1.0.2) to "DP approx likelihood 1-way" representing a much improved approach.

For the default configuration, it will also add some jitter noise to the data randomly only for clustering to avoid clustering failure by too many overlapping points in the data.

For more details, please check the documents.

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](https://doi.org/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 = x^2 # 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

#
```

gen_simdata

Generate Simulated Data

Description

Generate Simulated Data

Usage

```
gen_simdata(cord, sim_table, noise = 0.3, plot = FALSE)
```

Arguments

cord	data matrix that records the index for each cluster on each dimension
sim_table	a matrix
noise	a numeric value to specify noise level
plot	a logical to turn on or off plotting

hashed_upsilon	<i>Hashed Upsilon Test of Association</i>
----------------	---

Description

Hashed Upsilon Test of Association

Usage

```
hashed_upsilon(points, log.p = FALSE)
```

Arguments

points	a matrix of observed categorical data points. Rows are observations and columns are variables.
log.p	a logical. If TRUE, the returned p.value is logarithm of the true p-value. Useful for improved precision when true p-values are close to zero.

See Also

[upsilon.test](#) for full description of the Upsilon test.

plot.GridOnClusters	<i>Plotting the continuous data along with cluster-preserving Grid</i>
---------------------	--

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,
  col,
  line_col = "black",
  cex = 1.125,
  sub = NULL,
  pch = 19,
  plot.table = TRUE,
  ...
)
```

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
col	the color of data points
line_col	the color of grid lines
cex	A numerical value giving the amount by which plotting text and symbols should be magnified relative to the default.
sub	the subtitle
pch	the symbol for points on the scatter plots
plot.table	a logical to show the contingency table. Default: TRUE.
...	additional graphical parameters

plotGOCpatterns	<i>(OBOSOLETE) Plotting the continuous data along with cluster-preserving Grid</i>
-----------------	--

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 <code>discretize.jointly</code>

Index

ballclust, [2](#)

Ckmeans.1d.dp, [7](#)

cluster, [3](#)

discretize.jointly, [3](#)

gen_simdata, [7](#)

hashed_upsilon, [8](#)

plot.GridOnClusters, [8](#)

plotGOCpatterns, [9](#)

upsilon.test, [8](#)