Package ‘GridOnClusters’

December 12, 2025

Type Package

Title Multivariate Joint Grid Discretization

Version 0.3.2

Date 2025-12-12

Depends R (>= 3.5.0)

Author Jiandong Wang [aut],
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. It can handle both labeled or unlabeled data. Both published methods (Wang et al 2020) <doi:10.1145/3388440.3412415> and new methods are included. 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

RoxygenNote 7.3.3

NeedsCompilation yes

VignetteBuilder knitr

Repository CRAN

Date/Publication 2025-12-12 13:40:07 UTC

Contents

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

Description

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

Usage

```
discretize.jointly(
  data,
  k = c(2:10),
  min_level = 1,
  max_level = 100,
  cluster_method = c("Ball+BIC", "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", "upsllion", "CAIR"),
  cluster_label = NULL,
  cutoff = 0,
  entropy = FALSE,
  noise = FALSE,
  dim_reduction = FALSE,
  scale = FALSE,
  variance = 0.5,
  nthread = 1
)
```

Arguments

`data`	a numeric matrix for multivariate data or a numeric vector for univariate data. In case of a matrix, columns are continuous variables; rows are observations.
`k`	either an integer, a integer vector, or `Inf`, specifying the number of clusters. The default is a vector of 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 among the integers. If `k` is set to `Inf`, an optimal `k` is chosen from 2 to `nrow(data)`. If `cluster_label` is specified, `k` is ignored.

`min_level` an integer or an integer vector, to specify 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` an integer or an integer vector, to specify the maximum number of levels along each dimension. It works in the same way as `min_level`. `max_level` will be set to the smaller between number of compressed zones and itself, if `grid_method` is a likelihood approach or "DP Compressed majority".

`cluster_method` a character string to specify a clustering method to be used. Ignored if `cluster_label` is not `NULL`. We offer three build-in options:

"`Ball+BIC`" (default) uses `mclust::Mclust` (`modelNames = "VII"` for 2-D or higher dimensions; `"V"` for 1-D) to cluster `data` and BIC score to select number of clusters.

`"kmeans+silhouette"` uses k-means to cluster `data` and the average Silhouette width to select number of clusters.

`"PAM"` uses the algorithm partition around medoids to perform clustering.

`grid_method` a character string to specify a grid 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 a method to evaluate quality of discretized data.

`cluster_label` a vector of labels for each data point or observation. It can be class labels on the input `data` for supervised learning; it can also be cluster labels for unsupervised learning. If `NULL` (default), clustering is performed to obtain labels.

`cutoff` a numeric value. A grid line is added only when the quality of the line is not smaller than `cutoff`. It is applicable only to `grid_method` `"DP"` or `"DP Compressed majority"`.

`entropy` a logical to chose either entropy (`TRUE`) or likelihood (`FALSE`, default).

`noise` a logical to apply jitter noise to original data if `TRUE`. Default: `FALSE`. It is only applicable to `cluster_method` `"Ball+BIC"`. When data contain many duplicated values, adding noise can help `Mclust` clustering.

`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 if `TRUE`. Default: `FALSE`.

`variance` a numeric value to specify noise variance to be added to the data

`nthread` an integer to specify number of CPU threads to use. Automatically adjusted if invalid or exceeding available cores.

Details

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

The included grid discretization methods can be summarized into three categories:

- By Density

  - `"Sort+split"` (Wang et al. 2020) sorts clusters by mean in each dimension. It then splits consecutive pairs only if the sum of 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"` splits 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"` orders labels by data in each dimension and then cuts data into a maximum of `max_level` bins. It evaluates the quality of each cut to find a best number of bins.
  - `"DP Compressed majority"` orders labels by data in each dimension. It then compresses labels neighbored by the same label to avoid discretization within consecutive points of the same cluster label, so as to greatly reduce runtime of dynamic programming. Then it cuts data into a maximum of `max_level` bins, and it evaluates the quality of each cut by the majority of data to find a best number of bins.
- By cluster likelihood
  - `"DP exact likelihood"` orders labels by data in each dimension. It then compresses labels neighbored by the same label to avoid discretization within consecutive points of the same cluster label, so as to greatly reduce runtime of dynamic programming. Then cut the data into a maximum of `max_level` bins.
  - `"DP approx likelihood 1-way"` is a sped-up version of the `"DP exact likelihood"` method, but it is not always optimal.
  - `"DP approx likelihood 2-way"` is a bidirectional variant of the `"DP approx likelihood"` method. It performs approximate dynamic programming 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 not always achieved.

Value

A list that contains four items:

`D` a matrix of discretized values from original `data`. Discretized values are one(1)-based.

`grid` a list of numeric vectors of decision boundaries for each variable/dimension.

`clabels` a vector of 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 `grid_method` is changed from `"Sort+Split"` (Wang et al. 2020) (up to released package version 0.1.0.2) to `"DP approx likelihood 1-way"` (since version 0.3.2), representing a major improvement.

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 Grid on Continuous Data*

Description

Plots discretized data based on grid that preserves clusters in 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

Index