Package 'clusterlab'

January 22, 2019

Title Flexible Gaussian Cluster Simulator

Version 0.0.2.6

Date 2019-01-22

Author Christopher R John

Maintainer Christopher R John <chris.r.john86@gmail.com>

Depends R (>= 3.4.0)

Description

Clustering is a central task in big data analyses and clusters are often Gaussian or near Gaussian. However, a flexible Gaussian cluster simulation tool with precise control over the size, variance, and spacing of the clusters in NXN dimensional space does not exist. This is why we created 'clusterlab'. The algorithm first creates X points equally spaced on the circumference of a circle in 2D space. These form the centers of each cluster to be simulated. Additional samples are added by adding Gaussian noise to each cluster center and concatenating the new sample co-ordinates. Then if the feature space is greater than 2D, the generated points are considered principal component scores and projected into N dimensional space using linear combinations using fixed eigenvectors. Through using vector rotations and scalar multiplication clusterlab can generate complex patterns of Gaussian clusters and outliers.

License AGPL-3

Encoding UTF-8

LazyData true

Imports ggplot2, reshape

Suggests knitr

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-01-22 11:30:03 UTC

R topics documented:

clusterlab *clusterlab*

Description

This function runs clusterlab which is a simulator for Gaussian clusters. The default method positions cluster centers on the perimeter of a circle, before creating gaussian clusters around them and projecting the 2D co-ordinates into high dimensional feature space. This method allows control over the spacing, variance, and size of the clusters. Also included is a simple random cluster simulator where the spacing of the clusters cannot be controlled precisely, but the other parameters can.

Usage

```
clusterlab(centers = 1, r = 8, sdvec = NULL, alphas = NULL,
  centralcluster = FALSE, numbervec = NULL, features = 500,
  seed = NULL, rings = NULL, ringalphas = NULL, ringthetas = NULL,
  outliers = NULL, outlierdist = NULL, mode = c("circle", "random"),
  minalloweddist = 0, pcafontsize = 18, showplots = TRUE)
```

Arguments

centers	Numerical value: the number of clusters to simulate (N)
r	Numerical value: the number of units of the radius of the circle on which the clusters are generated
sdvec	Numerical vector: standard deviation of each cluster, N values are required
alphas	Numerical vector: how many units to push each cluster away from the initial placement, N values are required
centralcluster	Logical flag: whether to place a cluster in the middle of the rest
numbervec	Numerical vector: the number of samples in each cluster, N values are required
features	Numerical value: the number of features for the data
seed	Numerical value: fixes the seed if you want to repeat results, set the seed to 123 for example here
rings	Numerical value: the number of concentric rings to generate (previous settings apply to all ring clusters)
ringalphas	Numerical vector: a vector of numbers to push each ring out by, must equal number of rings
ringthetas	Numerical vector: a vector of angles to rotate each ring by, must equal number of rings
outliers	Numerical value: the number of outliers to create
outlierdist	Numerical value: a distance value to move the outliers by

mode	Character string: whether to use the standard method (circle), or simple random placement (random)
minalloweddist	Numerical value: minimum distance between the randomised cluster centers, otherwise repeat randomisation
pcafontsize	Numerical value: the font size of the pca
showplots	Logical flag: whether to remove the plots

Value

A list, containing: 1) the synthetic data 2) cluster membership matrix

Examples

```
synthetic <- clusterlab(centers=4,r=8,sdvec=c(2.5,2.5,2.5,2.5),
alphas=c(1,1,1,1),centralcluster=FALSE,
numbervec=c(50,50,50,50)) # for a six cluster solution)
```

Index