Package 'RidgeClust'

January 18, 2011

Type Package

Title A clustering algorithm based on the package "Mclust" and a merging approach. See ?ridgeline.clust

Version 1.0

Date 2010-12-17

Author Grigory Alexandrovich

Maintainer <alexandrovich@mathematik.uni-marburg.de>

Description A 2-step clustering-algorithm. At the 1. step a Mclust- (a Gaussian-Mixture-) clustering of the data is generated. At the 2. step components which lead to unimodal or weakly bimodal mixtures are merged. The measure of the bimodality is the ridgeline ratio. See ?ridgeline.clust.

License GPL-2

LazyLoad yes

Depends mclust, ellipse

R topics documented:

`plot.RidgeClust` *Plot function.*

Description

Redefined generic plot function for the class RidgeClust. Plots results of the clustering algorithm ridgeline.clust in the case of two-dimensional data. Shows the scatterplot of the original data, the result of Mclust and the result after the merging.

Usage

```
plot(x, daten)
```

Arguments

x	Result of the function ridgeline.clust().
daten	Data which was clustered (a matrix or a dataframe with two columns).

Author(s)

G. Alexandrovich

See Also

ridgeline.clust

Examples

```
#See example of ridgline.clust
```

RidgeClustPackage *A clustering algorithm based on Mclust and a merging approach. See ?ridgeline.clust*

Description

A 2-step clustering-algorithm. At the 1. step a Mclust- (a Gaussian-Mixture-) clustering of the data is generated. At the 2. step components which lead to unimodal or weakly bimodal mixtures are merged. The measure of the bimodality is the ridgeline ratio. See ?ridgeline.clust.

Details

Package:	ridgeline.clust
Type:	Package
Version:	1.0
Date:	2010-12-17
License:	GPL-2
LazyLoad:	yes

ridgeline.clust(data)

Author(s)

Grigory Alexandrovich,
alexandrovich@mathematik.uni-marburg.de

References

Alexandrovich G. (2010): Analytische Eigenschaften von Mischungen elliptischer Verteilungen und deren Anwendung in der Clusteranalyse, diploma thesis, Marburg University

Fraley, C., Raftery, A.E. (2006): Technical Report MCLUST Version 3 for R.

Hennig, C. (2009): Methods for merging Gaussian mixture components, Research Report no. 302 , Department of Statistical Science, UCL

Surajit Ray and Bruce G. Lindsay (2005): The topography of multivariate normal mixtures, Ann. Statist. 33, Number 5, pp. 2042-2065

See Also

`ridgeline.clust` `plot.RidgeClust`

`ridgeline.clust` *Computes a clustering of the data*

Description

A 2-step clustering-algorithm. At the 1. step a clustering based on a Gaussian mixture model is calculated for the data (the package Mclust is applied here). At the 2. step Gaussian components which lead to weakly seperated (i.e. unimodal or weakly bimodal) submixtures are merged, so that the number of clusters is reduced.

Usage

```
ridgeline.clust(daten, rst = 0.7, anz = 15, G = NULL, modelNames = NULL,
prior = NULL, control = emControl(), initialization = NULL, warn = FALSE, ...)
```

Arguments

`daten`	Data to be clustered, a dataframe or a matrix.
`rst`	Cutoff value for the ridgeline ratio (0 <= rst <= 1). rst = 1 implies merging only at a strict unimodality, rst near 0 implicates merging of strongly separated components.
`anz`	Parameter for the mode-searching routine. For high-dimensional data must be increased. A suitable choice is (15/2) * Dimension
`G`	Mclust parameter. Vector with numbers of components for the clustering. See Mclust reference
`modelNames`	Covariance structure of the components. See Mclust reference
`prior`	Priors. See Mclust reference
`control`	Parameter-Object for the EM-Algorithm. See Mclust reference
`initialization`	See Mclust reference
`warn`	See Mclust reference
`...`	See Mclust reference

Value

A list of 2 Objects.

`Mclust`	Result of the Mclust-algorithm. See Mclust reference.
`Merged`	Result of the merging-algorithm. It is a list with following components:
`Merged$klass`	Vector of cluster-lables for each datapoint
`Merged$info`	A kx3 matrix, where k is the number of merging iterations. i'th row of this matrix consists of the numbers of the merged components a iteration i and the ridgeline-ratio of this components. For example info[1,] = c(1,2,0.8) means that in the first iteration components 1 and 2 were merged and their ridgline-ratio was 0.8. The new component gets the number 1 (the lesser one)
`Merged$diecluster`	A list the i-th element of which contains the numbers of old-clusters which were merged and build now the i'th cluster.
`Merged$new_posteriors`	A N x M Matrix, where N is the number of datapoints and M is the number of clusters, which contains posteriori-probabilities, for each cluster and datapoint.

Author(s)

Grigory Alexandrovich

References

Alexandrovich G. (2010): Analytische Eigenschaften von Mischungen elliptischer Verteilungen und deren Anwendung in der Clusteranalyse, diploma thesis, Marburg University

Fraley, C., Raftery, A.E. (2006): Technical Report MCLUST Version 3 for R.

Hennig, C. (2009): Methods for merging Gaussian mixture components, Research Report no. 302 , Department of Statistical Science, UCL

Surajit Ray and Bruce G. Lindsay (2005): The topography of multivariate normal mixtures, Ann. Statist. 33, Number 5, pp. 2042-2065

See Also

`Mclust` `plot.RidgeClust`

Examples

```
library(mvtnorm)

 #Simulate 1000 points from a gaussian mixture of 5 components:
 #Defining the means
 mu_vek <-  array(dim=c(2,5))
 mu_vek[,1] <- c(4,5)
 mu_vek[,2] <- c(1.5,5)
 mu_vek[,3] <- c(2,4.5)
 mu_vek[,4] <- c(4.1,1)
 mu_vek[,5] <- c(5,1)

 #Defining the covariances
 sigma_vek <- array(dim=c(2,2,5))
```

```
sigma_vek[,,1] <- rbind(c(0.3,0.05),c(0.05,.3))
sigma_vek[,,2] <- rbind(c(0.1,0.05),c(0.05,.1))
sigma_vek[,,3] <- rbind(c(0.2,0),c(0,0.2))
sigma_vek[,,4] <- rbind(c(0.2,.1),c(.1,.2))
sigma_vek[,,5] <- rbind(c(.2,.1),c(.1,.2))

#Defining the weights
weight <- c(0.2,0.25,0.15,0.2,0.2)

#Simulation loop
ind <- 1
dat <- array(dim=c(1000,2))

while(ind <= 1000)
{
  component <- rmultinom(1,1,weight)
  component <- which(component == 1)
  dat[ind,] <- rmvnorm(1, mean = mu_vek[,component], sigma = sigma_vek[,,component])
  ind <- ind + 1
}

#Clustering
result <- ridgeline.clust(dat)

#Plot
plot(result,dat)
```

Index