Package 'BClustLonG'

December 15, 2017

Type Package

Title A Dirichlet Process Mixture Model for Clustering Longitudinal Gene Expression Data

Version 0.1.2

Author Jiehuan Sun [aut, cre], Jose D. Herazo-Maya[aut], Naftali Kaminski[aut], Hongyu Zhao [aut], and Joshua L. Warren [aut],

Maintainer Jiehuan Sun <jiehuan.sun@gmail.com>

Description Many clustering methods have been proposed, but most of them cannot work for longitudinal gene expression data. 'BClustLonG' is a package that allows us to perform clustering analysis for longitudinal gene expression data. It adopts a linear-mixed effects framework to model the trajectory of genes over time, while clustering is jointly conducted based on the regression coefficients obtained from all genes. To account for the correlations among genes and alleviate the high dimensionality challenges, factor analysis models are adopted for the regression coefficients. The Dirichlet process prior distribution is utilized for the means of the regression coefficients to induce clustering. This package allows users to specify which variables to use for clustering (intercepts or slopes or both) and whether a factor analysis model is desired. More details about this method can be found in Jiehuan Sun, et al. (2017) <doi:10.1002/sim.7374>.

License GPL-2

Encoding UTF-8

LazyData true

Depends R (>= 3.4.0), MASS (>= 7.3-47), lme4 (>= 1.1-13), mcclust (>= 1.0)

Imports Rcpp (>= 0.12.7)

Suggests knitr, lattice

VignetteBuilder knitr

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 6.0.1

NeedsCompilation yes

Repository CRAN
Date/Publication 2017-12-15 13:52:52 UTC

R topics documented:

`BClustLonG` *A Dirichlet process mixture model for clustering longitudinal gene expression data.*

Description

A Dirichlet process mixture model for clustering longitudinal gene expression data.

Usage

```
BClustLonG(data = NULL, iter = 20000, thin = 2, savePara = FALSE,
  infoVar = c("both", "int")[1], factor = TRUE, hyperPara = list(v1 = 0.1,
  v2 = 0.1, v = 1.5, c = 1, a = 0, b = 10, cd = 1, aa1 = 2, aa2 = 1, alpha0 =
  -1, alpha1 = -1e-04, cutoff = 1e-04, h = 100))
```

Arguments

`data`	Data list with three elements: Y (gene expression data with each column being one gene), ID, and years. (The names of the elements have to be matached exactly. See the data in the example section more info)
`iter`	Number of iterations (excluding the thinning).
`thin`	Number of thinnings.
`savePara`	Logical variable indicating if all the parameters needed to be saved. Default value is FALSE, in which case only the membership indicators are saved.
`infoVar`	Either "both" (using both intercepts and slopes for clustering) or "int" (using only intercepts for clustering)
`factor`	Logical variable indicating whether factor analysis model is wanted.
`hyperPara`	A list of hyperparameters with default values.

Value

returns a list with following objects.

`e.mat`	Membership indicators from all iterations.
`All other parameters`	only returned when savePara=TRUE.

References

Jiehuan Sun, Jose D. Herazo-Maya, Naftali Kaminski, Hongyu Zhao, and Joshua L. Warren. "A Dirichlet process mixture model for clustering longitudinal gene expression data." Statistics in Medicine 36, No. 22 (2017): 3495-3506.

Examples

```
data(data)
## increase the number of iterations
## to ensure convergence of the algorithm

res = BClustLonG(data, iter=20, thin=2,savePara=FALSE,
infoVar="both",factor=TRUE)
## discard the first 10 burn-ins in the e.mat
## and calculate similarity matrix
## the number of burn-ins has be chosen s.t. the algorithm is converged.
mat = calSim(t(res$e.mat[,11:20]))
clust = maxpear(mat)$cl ## the clustering results.
## Not run:
## if only want to include intercepts for clustering
## set infoVar="int"
res = BClustLonG(data, iter=10, thin=2,savePara=FALSE,
infoVar="int",factor=TRUE)

## if no factor analysis model is wanted
## set factor=FALSE
res = BClustLonG(data, iter=10, thin=2,savePara=FALSE,
infoVar="int",factor=TRUE)

## End(Not run)
```

`calSim` *Function to calculate the similarity matrix based on the cluster membership indicator of each iteration.*

Description

Function to calculate the similarity matrix based on the cluster membership indicator of each iteration.

Usage

```
calSim(mat)
```

Arguments

`mat`	Matrix of cluster membership indicator from all iterations

Examples

```
n = 90 ##number of subjects
iters = 200 ##number of iterations
## matrix of cluster membership indicators
## perfect clustering with three clusters
mat = matrix(rep(1:3,each=n/3),nrow=n,ncol=iters)
sim = calSim(t(mat))
```

---

data *Simulated dataset for testing the algorithm*

Description

Simulated dataset for testing the algorithm

Usage

```
data(data)
```

Examples

```
data(data)
## this is the required data input format
head(data.frame(ID=data$ID,years=data$years,data$Y))
```

Index