Package 'bpgmm'

August 2, 2019

Type Package

Title Bayesian Model Selection Approach for Parsimonious Gaussian Mixture Models

Version 1.0.5

Date 2019-07-15

Depends R(>= 3.1.0)

Imports methods (>= 3.5.1), mcmcse (>= 1.3-2), pgmm (>= 1.2.3), mvtnorm (>= 1.0-10), MASS (>= 7.3-51.1), Rcpp (>= 1.0.1), gtools (>= 3.8.1), label.switching (>= 1.8)

Author Xiang Lu <Xiang_Lu at urmc.rochester.edu>, Yaoxiang Li <yl814 at georgetown.edu>, Tanzy Love <tanzy_love at urmc.rochester.edu>

Maintainer Yaoxiang Li <yl814@georgetown.edu>

Description Model-based clustering using Bayesian parsimonious Gaussian mixture models. MCMC (Markov chain Monte Carlo) are used for parameter estimation. The RJMCMC (Reversible-jump Markov chain Monte Carlo) is used for model selection. GREEN et al. (1995) <doi:10.1093/biomet/82.4.711>.

SystemRequirements C++11

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests testthat

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-08-02 10:00:02 UTC

R topics documented:

`CalculateProposalLambda` *CalculateProposalLambda*

Description

CalculateProposalLambda

Usage

```
CalculateProposalLambda(hparam, thetaYList, CxyList, constraint, m, p,
  qVec)
```

Arguments

`hparam`	hparam
`thetaYList`	thetaYList
`CxyList`	CxyList
`constraint`	constraint
`m`	the number of clusters
`p`	the number of features
`qVec`	the vector of the number of factors in each clusters

Value

calculated lambda for proposal function

Examples

```
sample_data = "https://raw.githubusercontent.com/lzyacht/bpgmm-examples/master/data/sampleData.csv"
X = utils::read.table(sample_data, header = TRUE, sep = ',')
X = as.matrix(X)
nsim = 1
burn = 20
n = ncol(X)
p = nrow(X)
m = 2
qVec = rep(3,m)
qnew = 3
delta = 2
ggamma = 2
dVec = c(1,1,1)
sVec = c(1,1,1)
constraint = c(1,1,1)

hparam = new("Hparam", alpha1=1, alpha2=2, bbeta=3, delta=4, ggamma=5)
muBar = apply(X, MARGIN = 1, FUN = mean)
ZOneDim = kmeans(x = t(X), centers = m)$cluster
thetaYList = generatePriorThetaY(m, n, p, muBar, hparam, qVec, ZOneDim, constraint)


url <- paste0("https://github.com/lzyacht/bpgmm-examples/",
"blob/master/data/CalculateProposalLambda.RData?raw=true")
download.file(url, destfile= "CalculateProposalLambda.RData", mode = "wb")
load("CalculateProposalLambda.RData")
CalculateProposalLambda(hparam, thetaYList, CxyList, constraint, m, p, qVec)
```

`CalculateProposalPsy` *CalculateProposalPsy*

Description

CalculateProposalPsy

Usage

```
CalculateProposalPsy(hparam, thetaYList, CxyList, constraint, m, p, qVec)
```

Arguments

hparam	hparam
thetaYList	thetaYList
CxyList	CxyList
constraint	constraint
m	the number of clusters
p	the number of features
qVec	the vector of the number of factors in each clusters

Value

calculated psy for proposal function

---

calculateRatio *Log scale ratio calculation*

Description

Log scale ratio calculation

Usage

```
calculateRatio(deno, nume)
```

Arguments

deno	denominator.
nume	numerator.

Value

result of ratio

calculateVarList *calculateVarList*

Description

calculateVarList

Usage

```
calculateVarList(psyList, lambdaList)
```

Arguments

psyList	list of psy
lambdaList	list of lambda

Value

list of covariance matrix

evaluatePrior *evaluate Prior*

Description

evaluate prior value for parameter Theta and Y.

Usage

```
evaluatePrior(m, p, muBar, hparam, thetaYList, ZOneDim, qVec, constraint)
```

Arguments

m	m
p	p
muBar	mu_bar
hparam	hyper parameter class
thetaYList	theta Y list
ZOneDim	one dime of z
qVec	q vector
constraint	type of constraint

Value

evaluated prior value for parameter Theta and Y.

Examples

```
url <- paste0("https://github.com/lzyacht/bpgmm-examples/",
"blob/master/data/updatePostThetaY_example.RData?raw=true")
download.file(url, destfile= "updatePostThetaY_example.RData", mode = "wb")
load("updatePostThetaY_example.RData")
evaluatePrior(m, p, muBar,hparam, thetaYList, ZOneDim, qVec,constraint)
```

`evaluatePriorLambda` *evaluatePriorLambda*

Description

evaluate prior value for parameter Lambda

Usage

```
evaluatePriorLambda(p, m, alpha2, qVec, psy, lambda, constraint)
```

Arguments

`p`	the number of features
`m`	the number of clusters
`alpha2`	hyper parameter
`qVec`	the vector of the number of factors in each clusters
`psy`	parameter
`lambda`	parameter
`constraint`	the pgmm constraint, a vector of length three with binary entry. For example, c(1,1,1) means the fully constraint model

Value

evaluated prior value for parameter Lambda

evaluatePriorPsi *evaluatePriorPsi*

Description

evaluate prior value for parameter Psi

Usage

```
evaluatePriorPsi(psy, p, m, delta, bbeta, constraint)
```

Arguments

psy	parameter
p	the number of features
m	the number of clusters
delta	parameter
bbeta	parameter
constraint	parameter

Value

evaluated prior value for parameter Psi

EvaluateProposalLambda *EvaluateProposalLambda*

Description

EvaluateProposalLambda

Usage

```
EvaluateProposalLambda(hparam, thetaYList, CxyList, constraint, newlambda,
  m, qVec, p)
```

Arguments

hparam	hparam
thetaYList	thetaYList
CxyList	CxyList
constraint	constraint
newlambda	newlambda
m	the number of clusters
qVec	the vector of the number of factors in each clusters
p	the number of features

Value

density value from evaluating lambda

EvaluateProposalPsy *EvaluateProposalPsy*

Description

EvaluateProposalPsy

Usage

```
EvaluateProposalPsy(hparam, thetaYList, CxyList, constraint, newpsy, m, p,
  qVec)
```

Arguments

hparam	hparam
thetaYList	thetaYList
CxyList	CxyList
constraint	constraint
newpsy	newpsy
m	the number of clusters
p	the number of features
qVec	the vector of the number of factors in each clusters

Value

evaluation of proposal psy

generatePriorLambda *generatePriorLambda*

Description

evaluate prior value for parameter Lambda

Usage

```
generatePriorLambda(p, m, alpha2, qVec, psy, constraint)
```

Arguments

p	the number of features
m	the number of clusters
alpha2	hyper parameter
qVec	parameter
psy	parameter
constraint	parameter

Value

evaluated prior value for parameter Lambda

generatePriorPsi *generatePriorPsi*

Description

generate prior value for parameter Psi

Usage

```
generatePriorPsi(p, m, delta, bbeta, constraint)
```

Arguments

p	the number of features
m	the number of clusters
delta	hyperparameters
bbeta	hyperparameters
constraint	the pgmm constraint, a vector of length three with binary entry. For example, c(1,1,1) means the fully constraint model

Value

generated prior value for parameter Psi

`generatePriorThetaY` *PriorThetaY list*

Description

generate prior value for parameter Theta and Y.

Usage

```
generatePriorThetaY(m, n, p, muBar, hparam, qVec, ZOneDim, constraint)
```

Arguments

`m`	the number of cluster
`n`	sample size
`p`	number of covariates
`muBar`	parameter
`hparam`	hyperparameters
`qVec`	the vector of the number of factors in each clusters
`ZOneDim`	ZOneDim
`constraint`	constraint

Value

PriorThetaY list

Examples

```
sample_data = "https://raw.githubusercontent.com/lzyacht/bpgmm-examples/master/data/sampleData.csv"
X = utils::read.table(sample_data, header = TRUE, sep = ',')
X = as.matrix(X)
nsim = 1
burn = 20
n = ncol(X)
p = nrow(X)
m = 2
qVec = rep(3,m)
qnew = 3
delta = 2
ggamma = 2
dVec = c(1,1,1)
sVec = c(1,1,1)
```

```
constraint = c(1,1,1)
hparam = new("Hparam", alpha1=1, alpha2=2, bbeta=3, delta=4, ggamma=5)
muBar = apply(X, MARGIN = 1, FUN = mean)
ZOneDim = kmeans(x = t(X), centers = m)$cluster
generatePriorThetaY(m, n, p, muBar, hparam, qVec, ZOneDim, constraint)
```

`getmode` *getmode*

Description

getmode

Usage

```
getmode(v)
```

Arguments

`v`	vec of numeric values

Value

mode of the vec

`getZmat` *Tool for vector to matrix*

Description

Tool for vector to matrix

Usage

```
getZmat(ZOneDim, m, n)
```

Arguments

`ZOneDim`	a vector.
`m`	the number of cluster.
`n`	sample size.

Value

adjacency matrix

Hparam-class *An S4 class to represent a Hyper parameter.*

Description

An S4 class to represent a Hyper parameter.

Slots

`alpha1` A numeric value

`alpha2` A numeric value

`delta` A numeric value

`ggamma` A numeric value

`bbeta` A numeric value

Examples

```
new("Hparam", alpha1=1, alpha2=2, bbeta=3, delta=4, ggamma=5)
```

likelihood *likelihood*

Description

likelihood

Usage

```
likelihood(thetaYList, ZOneDim, qqVec, muBar, X)
```

Arguments

`thetaYList`	thetaYList
`ZOneDim`	ZOneDim
`qqVec`	qqVec
`muBar`	muBar
`X`	X

Value

likelihood value for parameters thetaYList

Examples

```
url <- paste0("https://github.com/lzyacht/bpgmm-examples/",
"blob/master/data/CalculateProposalLambda.RData?raw=true")


download.file(url, destfile= "CalculateProposalLambda.RData", mode = "wb")
load("CalculateProposalLambda.RData")
likelihood(thetaYList, ZOneDim,qVec,muBar, X)
```

---

listToStrVec *Convert list of string to vector of string*

Description

Convert list of string to vector of string

Usage

```
listToStrVec(stringList)
```

Arguments

stringList	list of string

Value

vector of string

---

parsimoniousGaussianMixtureModel *bpgmm Model-Based Clustering Using Baysian PGMM Carries out model-based clustering using parsimonious Gaussian mixture models. MCMC are used for parameter estimation. The RJMCMC is used for model selection.*

Description

bpgmm Model-Based Clustering Using Baysian PGMM Carries out model-based clustering using parsimonious Gaussian mixture models. MCMC are used for parameter estimation. The RJMCMC is used for model selection.

Usage

```
parsimoniousGaussianMixtureModel(niter, burn, X, n, p, delta, ggamma, m,
  qVec, qnew, constraint, dVec, sVec)
```

Arguments

niter	the number of iterations
burn	the number of burn in iterations
X	the observation matrix with size p * m
n	the number of observations
p	the number of features
delta	scaler hyperparameters
ggamma	scaler hyperparameters
m	the number of clusters
qVec	the vector of the number of factors in each clusters
qnew	the number of factor for a new cluster
constraint	the pgmm constraint, a vector of length three with binary entry. For example, c(1,1,1) means the fully constraint model
dVec	a vector of hyperparameters with length three, shape parameters for alpha1, alpha2 and bbeta respectively
sVec	a vector of hyperparameters with length three, rate parameters for alpha1, alpha2 and bbeta respectively

Value

parsimonious Gaussian mixture models classification results list

Examples

```
sample_data = "https://raw.githubusercontent.com/lzyacht/bpgmm-examples/master/data/sampleData.csv"
X = utils::read.table(sample_data, header = TRUE, sep = ',')
X = as.matrix(X)
nsim = 1
burn = 20
n = ncol(X)
p = nrow(X)
m = 2
qVec = rep(3,m)
qnew = 3
delta = 2
ggamma = 2
dVec = c(1,1,1)
sVec = c(1,1,1)
constraint = c(0,0,0)

parsimoniousGaussianMixtureModel(nsim,burn,X,n,p,delta,ggamma,m,qVec,qnew,constraint,dVec,sVec)
```

`sumerizeZ` *sumerizeZ*

Description

sumerizeZ

Usage

```
sumerizeZ(Zlist, index = 1:length(Zlist))
```

Arguments

`Zlist`	list of Z matrix
`index`	index of Zlist

Value

count mode of Z matrix

`ThetaYList` *ThetaYList-class*

Description

Definiton of ThetaYList parameter sets

Slots

`tao` A numeric vector

`psy` A list value

`M` A list value

`lambda` A list value

`Y` A list value

`updatePostThetaY` *Update posterior theta Y list*

Description

Update posterior theta Y list

Usage

```
updatePostThetaY(m, n, p, hparam, thetaYList, ZOneDim, qVec, constraint, X)
```

Arguments

`m`	the number of clusters.
`n`	the number of observations.
`p`	the number of features.
`hparam`	hyper parameters S4 class.
`thetaYList`	theta Y list
`ZOneDim`	ZOneDim
`qVec`	qVec
`constraint`	constraint
`X`	X

Value

ThetaYList S4 class

Examples

```
url <- paste0("https://github.com/lzyacht/bpgmm-examples/",
"blob/master/data/updatePostThetaY_example.RData?raw=true")
download.file(url, destfile= "updatePostThetaY_example.RData", mode = "wb")
load("updatePostThetaY_example.RData")
updatePostThetaY(m, n, p, hparam, thetaYList, ZOneDim, qVec, constraint, X)
```

Index