Package: BayesCPclust (via r-universe)

January 29, 2025

Title A Bayesian Approach for Clustering Constant-Wise Change-Point Data

Version 0.1.0

Description A Gibbs sampler algorithm was developed to estimate change points in constant-wise data sequences while performing clustering simultaneously. The algorithm is described in da Cruz, A. C. and de Souza, C. P. E ``A Bayesian Approach for Clustering Constant-wise Change-point Data" <doi:10.48550/arXiv.2305.17631>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports extraDistr, RcppAlgos, stats

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Ana Carolina da Cruz [aut, cre]

Maintainer Ana Carolina da Cruz <adacruz@uwo.ca>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2025-01-29 16:50:06 UTC

Config/pak/sysreqs libgmp3-dev

Contents

`data` *Error free data for all examples.*

Description

A dataset generated for examplification of Gibbs sampler using the model proposed in the paper "BayesCPclust: A Bayesian Approach for Clustering Constant-Wise Change-Point Data". The generation process is described in the paper with N = 5, M = 50, w = 10, d = 2, K = 2.

Usage

```
data
```

Format

A matrix with 50 rows and 5 columns

References

A.C. da Cruz, C.P.E. de Souza. "BayesCPclust: A Bayesian Approach for Clustering Constant-Wise Change-Point Data" arXiv, arXiv:2305.17631v3 .

data_a *Error free data for all examples.*

Description

A dataset generated for examplification of Gibbs sampler using the model proposed in the paper "BayesCPclust: A Bayesian Approach for Clustering Constant-Wise Change-Point Data". The generation process is described in the paper with N = 5, M = 50, w = 10, d = 2, K = 2.

Usage

```
data_a
```

Format

A list with three components: a matrix with 50 rows and 5 columns, a vector with the cluster assignments, a vector with variance components

References

A.C. da Cruz, C.P.E. de Souza. "BayesCPclust: A Bayesian Approach for Clustering Constant-Wise Change-Point Data" arXiv, arXiv:2305.17631v3 .

full_cond *Full conditional for lambda*

Description

Full conditional for lambda

Usage

```
full_cond(kstar, lambda, cluster, al, bl, K, N)
```

Arguments

kstar	A scalar with the number maximum of change points in all clusters
lambda	A scalar defining the parameter for the Truncate Poisson distribution that controls the number of change points (or its initial values)
cluster	A vector containing the cluster assignments for the data sequences (or its initial values)
al	The hyperparameter value for the shape parameter in the gamma prior for lambda
bl	The hyperparameter value for the scale parameter in the gamma prior for lambda
K	A vector containing the number of change points for each cluster (or its initial values)
N	A scalar representing the number of data sequences

Value

'full_cond' returns a numerical value corresponding to a sample from the full conditional for lambda

Note

This function is used within the Gibbs sampler, it is not expected to be used alone.

Examples

```
# Using hypothetical values to exemplification purposes
clusters <- c(1,1,2,1,2)
full_cond(kstar = 2, lambda = 3, cluster = clusters, al = 2, bl = 1000, K = c(2, 2), N = 5)
```

---

gibbs_alg *Gibbs sampler algorithm for simulated scenarios or real datasets*

Description

Gibbs sampler algorithm for simulated scenarios or real datasets

Usage

```
gibbs_alg(
  N,
  w,
  M,
  K,
  Tl,
  cluster,
  alpha,
  sigma2,
  bs = 1000,
  as = 2,
  al = 2,
  bl = 1000,
  a = 2,
  b = 1000,
  alpha0 = 1/100,
  kstar,
  lambda,
  Y,
  d,
  maxIter = 10000
)
```

Arguments

N	A scalar representing the number of observations
w	A scalar representing the minimum number of points in each interval between two change points
M	A scalar representing the number of points available for each observation
K	A vector containing the number of change points for each cluster (or its initial values)
Tl	A list containing a vector for each cluster determining the change-point positions in each cluster (or its initial values)
cluster	A vector containing the cluster assignments for the observations (or its initial values)
alpha	A list containing a vector for each cluster determining the constant level values for each interval between change points in each cluster (or its initial values)
sigma2	A vector with the variances of observations (or its initial values)
bs	The hyperparameter value for the scale parameter in the inverse-gamma prior for the variance component
as	The hyperparameter value for the shape parameter in the inverse-gamma prior for the variance component
al	The hyperparameter value for the shape parameter in the gamma prior for lambda
bl	The hyperparameter value for the scale parameter in the gamma prior for lambda
a	The hyperparameter value for the shape parameter in the gamma prior for alpha0
b	The hyperparameter value for the scale parameter in the gamma prior for alpha0
alpha0	A scalar defining the parameter for the Dirichlet process prior that controls the number of clusters (or its initial values)
kstar	A scalar with the number maximum of change points in all clusters
lambda	A scalar defining the parameter for the Truncate Poisson distribution that controls the number of change points (or its initial values)
Y	A matrix M x N with the data sequences
d	A scalar representing the number of clusters.
maxIter	A scalar for the number of iteration to run in the Gibbs sampler

Value

A list with each component representing the estimates for each iteration of the Gibbs sampler for each parameter

See Also

[run_gibbs()]

Examples

```
data(data)
# initial values for each paramter and each cluster
par.values <- list(K = c(0, 0), Tl = list(50, 50), alpha = list(5, 10))
#cluster assignment for each data sequence
cluster <- kmeans(t(data), 2)$cluster
# variance for each data sequence
sigma2 <- apply(data, 2, var)
res <- gibbs_alg(alpha0 = 1/100, N = 5, w = 10, M = 50, K = par.values$K,
Tl = par.values$Tl, cluster = cluster, alpha = par.values$alpha, sigma2 = sigma2,
bs = 1000, as = 2, al = 2, bl = 1000, a = 2, b = 1000, kstar = 2, lambda = 2,
Y = data, d = 2, maxIter = 10)
```

`logsumexp` *Transfor a vector with over- or underflow*

Description

Transfor a vector with over- or underflow

Usage

```
logsumexp(x, min_x = Inf)
```

Arguments

`x`	A vector with numbers
`min_x`	A numerical value to represent the minimum value to perform comparison with the actual minimum value of ‘x‘

Value

‘logsumexp‘ returns each element of the vector ‘x‘ transformed using the Log-Sum-Exp trick.

Examples

```
# Transforming all elements in a vector using the Log-Sum-Exp trick
x <- c(1, 2, 3, 4, 5, 6)
logsumexp(x)
```

Mode *Compute the mode of a numerical vector*

Description

Compute the mode of a numerical vector

Usage

```
Mode(x)
```

Arguments

x	A vector with numbers

Value

'Mode' returns a value representing the most frequent numerical value in the vector 'x'

Examples

```
# Finding the mode of a vector of numbers
x <- c(1, 2, 2, 3, 5, 8, 10)
Mode(x)
```

pk *Probability mass function for truncated poisson*

Description

Probability mass function for truncated poisson

Usage

```
pk(k, kstar, lambda)
```

Arguments

k	A scalar for the number of changes points in a cluster
kstar	A scalar with the number maximum of change points in all clusters
lambda	A scalar defining the parameter for the Truncate Poisson distribution that controls the number of change points (or its initial values)

Value

'pk' returns a numerical value representing the marginal probability for a given k

Note

This function is used within the Gibbs sampler, it is not expected to be used alone.

See Also

[gibbs_alg()]

Examples

```
# Hypothetical values
pk(k = 2, kstar = 3, lambda = 2)
```

`possigma2n` *Full conditional function for sigma2*

Description

Full conditional function for sigma2

Usage

```
possigma2n(as, bs, M, Yn, k, Tln, alphan)
```

Arguments

`as`	The hyperparameter value for the shape parameter in the inverse-gamma prior for the variance component
`bs`	The hyperparameter value for the scale parameter in the inverse-gamma prior for the variance component
`M`	A scalar representing the number of points available for each data sequence
`Yn`	A vector or matrix with data sequences for a cluster
`k`	A scalar for the number of changes points in a cluster
`Tln`	A vector with the change-point positions for a cluster
`alphan`	A vector with the constant level values for each interval between change points for a cluster

Value

A numerical value corresponding to a sampled value from the full conditional of the variance component

Note

This function is called within the Gibbs sampler, but it can be used separately as well.

See Also

[gibbs_alg()]

Examples

```
data(data)
possigma2n(as = 2, bs = 1000, M = 50, Yn = data[,1], k = 0, Tln = 50, alphan = 15)
```

`postalpha0` *Posterior for alpha0*

Description

Posterior for alpha0

Usage

```
postalpha0(alpha0, a, b, N, cluster)
```

Arguments

`alpha0`	A scalar defining the parameter for the Dirichlet process prior that controls the number of clusters (or its initial values)
`a`	The hyperparameter value for the shape parameter in the gamma prior for alpha0
`b`	The hyperparameter value for the scale parameter in the gamma prior for alpha0
`N`	A scalar representing the number of data sequences
`cluster`	A vector containing the cluster assignments for the data sequences (or its initial values)

Value

A numerical value corresponding to a sample from the posterior of alpha0

Note

This function is called within the Gibbs sampler, but it can be called seperately.

Examples

```
postalpha0(alpha0 = 1/100, a = 2, b = 1000, N = 5, cluster = c(1,1,2,1,1))
```

postalphak *Full conditional for alphak*

Description

Full conditional for alphak

Usage

```
postalphak(M, Y, sigma2, K, Tl, cluster, clusteri)
```

Arguments

M	A scalar representing the number of points available for each data sequence
Y	A matrix M x N with the data sequences
sigma2	A vector with the variances of the data sequences (or its initial values)
K	A vector containing the number of change points for each cluster (or its initial values)
Tl	A list containing a vector for each cluster determining the change-point positions in each cluster (or its initial values)
cluster	A vector containing the cluster assignments for the data sequences (or its initial values)
clusteri	A scalar with the index of a cluster

Value

A numerical vector of size ‘K‘ + 1 with sampled values from the full conditional of alphak for a given cluster ‘clusteri‘

Note

This function is called within the Gibbs sampler, but it can be called separately as well.

See Also

[gibbs_alg()]

Examples

```
data(data)
postalphak(M = 50, Y = data, sigma2 = 0.05, K = c(0, 0), Tl = c(50, 50),
 cluster = c(1,1,2,1,2), clusteri = 1)
```

postK *Marginal probability of K*

Description

Marginal probability of K

Usage

```
postK(kstar, w, M, Y, cluster, sigma2, lambda, clusteri)
```

Arguments

kstar	A scalar with the number maximum of change points in all clusters
w	A scalar representing the minimum number of points in each interval between two change points
M	A scalar representing the number of points available for each data sequence
Y	A matrix M x N with the data sequences
cluster	A vector containing the cluster assignments for the data sequences (or its initial values)
sigma2	A vector with the variances of the data sequences (or its initial values)
lambda	A scalar defining the parameter for the Truncate Poisson distribution that controls the number of change points (or its initial values)
clusteri	A scalar with the index of a cluster

Value

A numerical value corresponding to the sampled number of change points, k, for a given cluster

Note

This function is called within the Gibbs sampler, but it can also de called separately.

See Also

[gibbs_alg()]

Examples

```
postK(kstar = 2, w = 10, M = 50, Y = data, cluster = c(1,1,2,1,2),
sigma2 = apply(data, 2, var), lambda = 2, clusteri = 1)
```

postK_mk *Marginal probability of K per bin*

Description

Marginal probability of K per bin

Usage

```
postK_mk(k, m0, w, M, Yn, sigma2n, cellsn, mk, Cr)
```

Arguments

k	A scalar for the number of changes points in a cluster
m0	A scalar for the number of positions available to define change-points positions
w	A scalar representing the minimum number of points in each interval between two change points
M	A scalar representing the number of points available for each data sequence
Yn	A vector or matrix with data sequences for a cluster
sigma2n	A vector with the variance of the data sequences in a cluster
cellsn	A vector with the indices of the data sequences in a cluster
mk	A matrix with all possible values to distribute between change points
Cr	A scalar with the number of data sequences in a cluster

Value

'postK_mk' returns a numerical value representing the non-normalized probability for a given bin, given k, and a given cluster

Note

This function is called within [postK()]. It should not be called alone.

See Also

[postK()], [gibbs_alg()]

Examples

```
data(data)
M <- 50; k <- 0; w <- 10;
m0 <- M - 1 -(k+1)*w
for(k in 0:2){
mk <- RcppAlgos::permuteGeneral(0:m0, k + 1,
constraintFun = "sum",
comparisonFun = "==", limitConstraints = m0,
```

```
repetition = TRUE)}
out <- postK_mk(k = 0, m0 = m0, w = 10, M = 50, Yn = data[,c(1,2,4)],
 sigma2n = rep(0.05, 3), cellsn = c(1,2,4), mk = mk[1,], Cr = 3)
```

---

postmk *Marginal probability of m1,m2,m3,...,mk+1*

Description

Marginal probability of m1,m2,m3,...,mk+1

Usage

```
postmk(w, M, Y, K, cluster, sigma2, clusteri)
```

Arguments

w	A scalar representing the minimum number of points in each interval between two change points
M	A scalar representing the number of points available for each data sequence
Y	A matrix M x N with the data sequences
K	A vector containing the number of change points for each cluster (or its initial values)
cluster	A vector containing the cluster assignments for the data sequences (or its initial values)
sigma2	A vector with the variances of the data sequences (or its initial values)
clusteri	A scalar with the index of a cluster

Value

A numerical vector of size k + 1 with the sampled number of observations (or bin size, mk) between each change point for a given cluster

Note

This function is called within the Gibbs sampler, but it can also be called separately.

Examples

```
data(data)
postmk(w = 10, M = 50, Y = data, K = c(1, 1), cluster = c(2,1,1,1,1), sigma2 = apply(data, 2, var),
clusteri = 1)
```

qn0 *Mixing probability for creating new cluster*

Description

Mixing probability for creating new cluster

Usage

```
qn0(alpha0, w, N, M, bs, as, kstar, lambda, Yn)
```

Arguments

alpha0	A scalar defining the parameter for the Dirichlet process prior that controls the number of clusters (or its initial values)
w	A scalar representing the minimum number of points in each interval between two change points
N	A scalar representing the number of data sequences
M	A scalar representing the number of points available for each data sequence
bs	The hyperparameter value for the scale parameter in the inverse-gamma prior for the variance component
as	The hyperparameter value for the shape parameter in the inverse-gamma prior for the variance component
kstar	A scalar with the number maximum of change points in all clusters
lambda	A scalar defining the parameter for the Truncate Poisson distribution that controls the number of change points (or its initial values)
Yn	A vector or matrix with data sequences for a cluster

Value

A numerical value representing the mixing value term used to compute the probability that the given data sequence should be a singleton cluster

Note

This function is called within [gibbs_alg()]. It should not be called alone.

See Also

[gibbs_alg()]

Examples

```
qn0(alpha0 = 1/100, w = 10, N = 5, M = 50, bs = 1000, as = 2, kstar = 2, lambda = 2, Yn = data[,1])
```

qn0_mk *Mixing probability for creating new cluster per bin*

Description

Mixing probability for creating new cluster per bin

Usage

```
qn0_mk(w, m0, bs, as, M, km, lambda, mk, Yn, kstar)
```

Arguments

w	A scalar representing the minimum number of points in each interval between two change points
m0	A scalar for the number of positions available to define change-points positions
bs	The hyperparameter value for the scale parameter in the inverse-gamma prior for the variance component
as	The hyperparameter value for the shape parameter in the inverse-gamma prior for the variance component
M	A scalar representing the number of points available for each data sequence
km	A scalar for the number of changes points in a cluster
lambda	A scalar defining the parameter for the Truncate Poisson distribution that controls the number of change points (or its initial values)
mk	A matrix with all possible values to distribute between change points
Yn	A vector with a data sequence
kstar	A scalar with the number maximum of change points in all clusters

Value

A numerical value representing the mixing value term used to compute the probability that the given data sequence should be a singleton cluster for a given bin size.

Note

This function is called within [qn0()]. It should not be called alone.

See Also

[qn0()], [gibbs_alg()]

Examples

```
data(data)
M <- 50; k <- 0; w <- 10;
m0 <- M - 1 -(k+1)*w
for(k in 0:2){
mk <- RcppAlgos::permuteGeneral(0:m0, k + 1,
constraintFun = "sum",
comparisonFun = "==", limitConstraints = m0,
repetition = TRUE)}
out <- qn0_mk(w = 10, m0 = m0, bs = 1000, as = 2, M = 50, km = 1,
 lambda = 2, mk = mk[1,], Yn = data[,1], kstar = 2)
```

---

qnj *Mixing probability for getting assigned to an existing cluster*

Description

Mixing probability for getting assigned to an existing cluster

Usage

```
qnj(N, M, as, bs, Yn, alpha, cluster, Tl, K)
```

Arguments

N	A scalar representing the number of data sequences
M	A scalar representing the number of points available for each data sequence
as	The hyperparameter value for the shape parameter in the inverse-gamma prior for the variance component
bs	The hyperparameter value for the scale parameter in the inverse-gamma prior for the variance component
Yn	A vector or matrix with data sequences for a cluster
alpha	A list containing a vector for each cluster determining the constant level values for each interval between change points in each cluster (or its initial values)
cluster	A vector containing the cluster assignments for the data sequences (or its initial values)
Tl	A list containing a vector for each cluster determining the change-point positions in each cluster (or its initial values)
K	A vector containing the number of change points for each cluster (or its initial values)

Value

A vector of same size as the vector ‘cluster‘ corresponding to the mixing term value used to compute the probability that the given data sequence ‘Yn‘ should be part of each existing cluster

Note

This function is called within the Gibbs sampler. It should not be called alone.

See Also

[gibbs_alg()]

Examples

```
qnj(N = 5, M = 50, as = 2, bs = 1000, Yn = data[,1], alpha = c(10, 10),
 cluster = c(1,1,2,1,2), Tl = c(50,50), K = c(0,0))
```

`run_gibbs` *Runs the Gibbs sampler algorithm using using initial values for the parameters*

Description

Runs the Gibbs sampler algorithm using using initial values for the parameters

Usage

```
run_gibbs(
  M,
  N,
  w,
  d,
  as = 2,
  bs = 100,
  al = 2,
  bl = 1000,
  a = 2,
  b = 1000,
  alpha0 = 1/100,
  lambda = 2,
  maxIter = 10000,
  par.values,
  data,
  cluster,
  sigma2
)
```

Arguments

M	A scalar representing the number of points available for each observation
N	A scalar representing the number of observations
w	A scalar representing the minimum number of points in each interval between two change points
d	A scalar representing the number of clusters.
as	The hyperparameter value for the shape parameter in the inverse-gamma prior for the variance component
bs	The hyperparameter value for the scale parameter in the inverse-gamma prior for the variance component
al	The hyperparameter value for the shape parameter in the gamma prior for lambda
bl	The hyperparameter value for the scale parameter in the gamma prior for lambda
a	The hyperparameter value for the shape parameter in the gamma prior for alpha0
b	The hyperparameter value for the scale parameter in the gamma prior for alpha0
alpha0	A scalar defining the parameter for the Dirichlet process prior that controls the number of clusters (or its initial values)
lambda	A scalar defining the parameter for the Truncate Poisson distribution that controls the number of change points (or its initial values)
maxIter	A scalar for the number of iteration to run in the Gibbs sampler
par.values	A list with lists with parameters for each cluster. The first argument in each list is the number of change points, then the positions for the change points, where T_1 = 1, T_last = M + 1, and for each interval between change points you need to specify a value for the constant level. If running the Gibbs sampler for a dataset with unknown number of change points, we suggest setting the number of change points for each cluster to be zero. Check example in README file.
data	a matrix of size M x N with data sequences in the columns
cluster	a vector with cluster assignments for each data sequence
sigma2	a vector with variance components for each data sequence

Value

A list with estimates for each iteration of the Gibbs sampler for each parameter

Examples

```
d = 2 # two clusters
N = 5 # 5 data sequences
M = 50 # 50 observations for each data sequence
maxIter = 10 # number of Gibbs sampler iterations

data(data)
# initial values for each paramter and each cluster
par.values <- list(K = c(0, 0), Tl = list(50, 50), alpha = list(5, 10))
#cluster assignment for each data sequence
```

```
cluster <- kmeans(t(data), 2)$cluster
# variance for each data sequence
sigma2 <- apply(data, 2, var)
res <- run_gibbs(M, N, w = 10, d, as = 2, bs = 100, al = 2, bl = 1000, a = 2,
 b = 1000, alpha0 = 1/100, lambda = 2, maxIter = 10, par.values, data,
 cluster, sigma2)
```

`update_lambda` *Update equation for lambda*

Description

Update equation for lambda

Usage

```
update_lambda(a = 4, b = 2, kstar, lambda, cluster, al, bl, K, N)
```

Arguments

`a`	The hyperparameter value for the shape parameter in the gamma prior for alpha0
`b`	The hyperparameter value for the scale parameter in the gamma prior for alpha0
`kstar`	A scalar with the number maximum of change points in all clusters
`lambda`	A scalar defining the parameter for the Truncate Poisson distribution that controls the number of change points (or its initial values)
`cluster`	A vector containing the cluster assignments for the data sequences (or its initial values)
`al`	The hyperparameter value for the shape parameter in the gamma prior for lambda
`bl`	The hyperparameter value for the scale parameter in the gamma prior for lambda
`K`	A vector containing the number of change points for each cluster (or its initial values)
`N`	A scalar representing the number of data sequences

Value

A numerical value corresponding to a sample from the posterior of the parameter lambda

Note

This function is called within the Gibbs sampler, but it can also be called separately.

See Also

[gibbs_alg()]

Examples

```
update_lambda(a = 4, b = 2, kstar = 2, lambda = 2, cluster = c(1,1,2,1,2),
 al = 2, bl = 1000, K = c(2,2), N = 5)
```

Index