Package 'SANple'

June 2, 2024

Type Package

Title Fitting Shared Atoms Nested Models via Markov Chains Monte Carlo

Version 0.1.1

Date 2024-06-02

Maintainer Francesco Denti <francescodenti.personal@gmail.com>

URL <https://github.com/laura-dangelo/SANple>

BugReports <https://github.com/laura-dangelo/SANple/issues>

Description Estimate Bayesian nested mixture models via Markov Chain Monte Carlo methods. Specifically, the package implements the common atoms model (Denti et al., 2023), its finite version (D'Angelo et al., 2023), and a hybrid finite-infinite model.
All models use Gaussian mixtures with a normal-inverse-gamma prior distribution on the parameters. Additional functions are provided to help analyzing the results of the fitting procedure.
References:
Denti, Camerlenghi, Guindani, Mira (2023) <doi:10.1080/01621459.2021.1933499>,
D'Angelo, Canale, Yu, Guindani (2023) <doi:10.1111/biom.13626>.

License MIT + file LICENSE

Imports Rcpp, salso

Depends scales, RColorBrewer

LinkingTo Rcpp, RcppArmadillo, RcppProgress

RoxygenNote 7.3.1

Encoding UTF-8

NeedsCompilation yes

Author Francesco Denti [aut, cre] (<https://orcid.org/0000-0003-2978-4702>),
Laura D'Angelo [aut, cph] (<https://orcid.org/0000-0001-5034-7414>)

Repository CRAN

Date/Publication 2024-06-02 13:00:02 UTC

R topics documented:

---

estimate_clusters   *Estimate observational and distributional clusters*

Description

Given the MCMC output, estimate the observational and distributional partitions using salso::salso().

Usage

```
estimate_clusters(object, burnin = 0, ncores = 0)
```

Arguments

object	object of class SANmcmc (the result of a call to sample_fiSAN, sample_fiSAN_sparsemix, sample_fSAN, sample_fSAN_sparsemix, or sample_CAM).
burnin	the length of the burn-in to be discarded before estimating the clusters (default is 2/3 of the iterations).
ncores	the number of CPU cores to use, i.e., the number of simultaneous runs at any given time. A value of zero indicates to use all cores on the system.

Value

Object of class SANclusters. The object contains:

est_oc estimated partition at the observational level. It is an object of class salso.estimate.

est_dc estimated partition at the distributional level. It is an object of class salso.estimate.

clus_means cluster-specific sample means of the estimated partition.

clus_vars cluster-specific sample variances of the estimated partition.

See Also

salso::salso(), print.SANmcmc, plot.SANmcmc, print.SANclusters

Examples

```
set.seed(123)
y <- c(rnorm(40,0,0.3), rnorm(20,5,0.3))
g <- c(rep(1,30), rep(2, 30))
out <- sample_fiSAN(nrep = 500, burn = 200,
                      y = y, group = g,
                    nclus_start = 2,
                    maxK = 20, maxL = 20,
                    beta = 1)
estimate_clusters(out)
```

---

plot.SANmcmc   *Plotting MCMC output*

Description

Plot method for objects of class SANmcmc. The function displays two graphs, meant to analyze the estimated distributional and observational clusters.

Usage

```
## S3 method for class 'SANmcmc'
plot(
  x,
  type = c("boxplot", "ecdf", "scatter"),
  estimated_clusters = NULL,
  burnin = 0,
  palette_brewed = FALSE,
  ncores = 1,
  ...
)
```

Arguments

x	object of class SANmcmc (the result of a call to sample_fiSAN, sample_fiSAN_sparsemix, sample_fSAN, sample_fSAN_sparsemix, or sample_CAM).
type	what type of plot should be drawn (only for the left-side plot). Possible types are "boxplot", "ecdf", and "scatter".
estimated_clusters	
	the output of a call to estimate_clusters (optional). It can be used to speed up the function if the partition has already been computed. If estimated_clusters = NULL, the displayed partition is computed using estimate_clusters.
burnin	the length of the burn-in to be discarded (default is 2/3 of the iterations).
palette_brewed	(logical) the color palette to be used. Default is R base colors (palette_brewed = FALSE).

ncores	if the partition is computed, the number of CPU cores to use to estimate the clusters, i.e., the number of simultaneous runs at any given time. A value of zero indicates to use all cores on the system.
...	additional graphical parameters to be passed when type = "scatter" is used.

Value

The function plots a summary of the fitted model.

See Also

[print.SANmcmc](#), [estimate_clusters](#)

Examples

```
set.seed(123)
y <- c(rnorm(40,0,0.3), rnorm(20,5,0.3))
g <- c(rep(1,30), rep(2, 30))
out <- sample_fiSAN(nrep = 500, burn = 200,
                    y = y, group = g,
                    nclus_start = 2,
                    maxK = 20, maxL = 20,
                    beta = 1)
plot(out, type = "ecdf", palette_brewed = TRUE)
```

---

print.SANclusters *Print cluster summary*

Description

Print the cluster-specific sample means and variances of the estimated observational and distributional partition.

Usage

```
## S3 method for class 'SANclusters'
print(x, ...)
```

Arguments

x	object of class SANclusters (the result of a call to [estimate_clusters](#))
...	ignored.

Value

The function prints a summary of the estimated clusters.

---

print.SANmcmc   *Print MCMC output*

Description

Print method for objects of class SANmcmc.

Usage

```
## S3 method for class 'SANmcmc'
print(x, ...)
```

Arguments

x   object of class SANmcmc (the result of a call to sample_fiSAN, sample_fiSAN_sparsemix,
                sample_fSAN, sample_fSAN_sparsemix, or sample_CAM).

...   ignored.

Value

The function prints a summary of the fitted model.

See Also

estimate_clusters, plot.SANmcmc

---

sample_CAM   *Sample CAM*

Description

sample_CAM is used to perform posterior inference under the common atoms model (CAM) of Denti
et al. (2023) with Gaussian likelihood. The model uses Dirichlet process mixtures (DPM) at both
the observational and distributional levels. The implemented algorithm is based on the nested slice
sampler of Denti et al. (2023), based on the algorithm of Kalli, Griffin and Walker (2011).

Usage

```
sample_CAM(nrep, burn, y, group,
           maxK = 50, maxL = 50,
           m0 = 0, tau0 = 0.1, lambda0 = 3, gamma0 = 2,
           hyp_alpha1 = 1, hyp_alpha2 = 1,
           hyp_beta1 = 1, hyp_beta2 = 1,
           alpha = NULL, beta = NULL,
           warmstart = TRUE, nclus_start = NULL,
```

```
             mu_start = NULL, sigma2_start = NULL,
             M_start = NULL, S_start = NULL,
             alpha_start = NULL, beta_start = NULL,
             progress = TRUE, seed = NULL)
```

Arguments

nrep	Number of MCMC iterations.
burn	Number of discarded iterations.
y	Vector of observations.
group	Vector of the same length of y indicating the group membership (numeric).
maxK	Maximum number of distributional clusters (default = 50).
maxL	Maximum number of observational clusters (default = 50).
m0, tau0, lambda0, gamma0	Hyperparameters on $(\mu, \sigma^2) \sim NIG(m_0, \tau_0, \lambda_0, \gamma_0)$. Default is (0, 0.1, 3, 2).
hyp_alpha1, hyp_alpha2	If a random α is used, (hyp_alpha1, hyp_alpha2) specify the hyperparameters. Default is (1,1). The prior is α ~ Gamma(hyp_alpha1, hyp_alpha2).
hyp_beta1, hyp_beta2	If a random β is used, (hyp_beta1, hyp_beta2) specify the hyperparameters. Default is (1,1). The prior is β ~ Gamma(hyp_beta1, hyp_beta2).
alpha	Distributional DP parameter if fixed (optional). The distribution is $\pi \sim GEM(\alpha)$.
beta	Observational DP parameter if fixed (optional). The distribution is $\omega_k \sim GEM(\beta)$.
warmstart, nclus_start	Initialization of the observational clustering. warmstart is logical parameter (default = TRUE) of whether a kmeans clustering should be used to initialize the chains. An initial guess of the number of observational clusters can be passed via the nclus_start parameter (optional). Default is nclus_start = min(c(maxL, 30)).
mu_start, sigma2_start, M_start, S_start, alpha_start, beta_start	Starting points of the MCMC chains (optional). mu_start, sigma2_start are vectors of length maxL. M_start is a vector of observational cluster allocation of length N. S_start is a vector of observational cluster allocation of length J. alpha_start, alpha_start are numeric.
progress	show a progress bar? (logical, default TRUE).
seed	set a fixed seed.

Details

Data structure

The common atoms mixture model is used to perform inference in nested settings, where the data are organized into J groups. The data should be continuous observations $(Y_1, \ldots, Y_J)$, where each $Y_j = (y_{1,j}, \ldots, y_{n_j,j})$ contains the n_j observations from group j, for $j = 1, \ldots, J$. The function takes as input the data as a numeric vector y in this concatenated form. Hence y should be a vector of length $n_1 + \cdots + n_J$. The group parameter is a numeric vector of the same size as y

indicating the group membership for each individual observation. Notice that with this specification the observations in the same group need not be contiguous as long as the correspondence between the variables y and group is maintained.

Model

The data are modeled using a univariate Gaussian likelihood, where both the mean and the variance are observational-cluster-specific, i.e.,

$$y_{i,j} \mid M_{i,j} = l \sim N(\mu_l, \sigma_l^2)$$

where $M_{i,j} \in \{1, 2, \dots\}$ is the observational cluster indicator of observation i in group j. The prior on the model parameters is a Normal-Inverse-Gamma distribution $(\mu_l, \sigma_l^2) \sim NIG(m_0, \tau_0, \lambda_0, \gamma_0)$, i.e., $\mu_l \mid \sigma_l^2 \sim N(m_0, \sigma_l^2/\tau_0)$, $1/\sigma_l^2 \sim Gamma(\lambda_0, \gamma_0)$ (shape, rate).

Clustering

The model performs a clustering of both observations and groups. The clustering of groups (distributional clustering) is provided by the allocation variables $S_j \in \{1, 2, \dots\}$, with

$$Pr(S_j = k \mid \dots) = \pi_k \qquad \text{for } k = 1, 2, \dots$$

The distribution of the probabilities is $\{\pi_k\}_{k=1}^\infty \sim GEM(\alpha)$, where GEM is the Griffiths-Engen-McCloskey distribution of parameter α, which characterizes the stick-breaking construction of the DP (Sethuraman, 1994).

The clustering of observations (observational clustering) is provided by the allocation variables $M_{i,j} \in \{1, 2, \dots\}$, with

$$Pr(M_{i,j} = l \mid S_j = k, \dots) = \omega_{l,k} \qquad \text{for } k = 1, 2, \dots; \ l = 1, 2, \dots$$

The distribution of the probabilities is $\{\omega_{l,k}\}_{l=1}^\infty \sim GEM(\beta)$ for all $k = 1, 2, \dots$

Value

sample_CAM returns four objects:

- model: name of the fitted model.
- params: list containing the data and the parameters used in the simulation. Details below.
- sim: list containing the simulated values (MCMC chains). Details below.
- time: total computation time.

Data and parameters: params is a list with the following components:

nrep Number of MCMC iterations.

y, group Data and group vectors.

maxK, maxL Maximum number of distributional and observational clusters.

m0, tau0, lambda0, gamma0 Model hyperparameters.

(hyp_alpha1,hyp_alpha2) **or** alpha Either the hyperparameters on α (if α random), or the value for α (if fixed).

(hyp_beta1,hyp_beta2) **or** beta Either the hyperparameters on β (if β random), or the value for β (if fixed).

Simulated values: `sim` is a list with the following components:

mu Matrix of size (`nrep`, `maxL`). Each row is a posterior sample of the mean parameter for each observational cluster $(\mu_1, \ldots \mu_L)$.

sigma2 Matrix of size (`nrep`, `maxL`). Each row is a posterior sample of the variance parameter for each observational cluster $(\sigma_1^2, \ldots \sigma_L^2)$.

obs_cluster Matrix of size (`nrep`, n), with n = `length(y)`. Each row is a posterior sample of the observational cluster allocation variables $(M_{1,1}, \ldots, M_{n_J,J})$.

distr_cluster Matrix of size (`nrep`, J), with J = `length(unique(group))`. Each row is a posterior sample of the distributional cluster allocation variables $(S_1, \ldots, S_J)$.

pi Matrix of size (`nrep`, `maxK`). Each row is a posterior sample of the distributional cluster probabilities $(\pi_1, \ldots, \pi_{maxK})$.

omega 3-d array of size (`maxL`, `maxK`, `nrep`). Each slice is a posterior sample of the observational cluster probabilities. In each slice, each column k is a vector (of length `maxL`) observational cluster probabilities $(\omega_{1,k}, \ldots, \omega_{maxL,k})$ for distributional cluster k.

alpha Vector of length `nrep` of posterior samples of the parameter α.

beta Vector of length `nrep` of posterior samples of the parameter β.

maxK Vector of length `nrep` of the number of distributional DP components used by the slice sampler.

maxL Vector of length `nrep` of the number of observational DP components used by the slice sampler.

References

Denti, F., Camerlenghi, F., Guindani, M., and Mira, A. (2023). A Common Atoms Model for the Bayesian Nonparametric Analysis of Nested Data. *Journal of the American Statistical Association*, 118(541), 405-416. <doi:10.1080/01621459.2021.1933499>

Kalli, M., Griffin, J.E., and Walker, S.G. (2011). Slice Sampling Mixture Models, *Statistics and Computing*, 21, 93–105. <doi:10.1007/s11222-009-9150-y>

Sethuraman, A.J. (1994). A Constructive Definition of Dirichlet Priors, *Statistica Sinica*, 4, 639–650.

Examples

```
set.seed(123)
y <- c(rnorm(40,0,0.3), rnorm(20,5,0.3))
g <- c(rep(1,30), rep(2, 30))
plot(density(y[g==1]), xlim = c(-5,10))
lines(density(y[g==2]), col = 2)
out <- sample_CAM(nrep = 500, burn = 200, y = y, group = g,
                  nclus_start = 2,
                  maxL = 20, maxK = 20)
out
```

---

sample_fiSAN   *Sample fiSAN*

Description

`sample_fiSAN` is used to perform posterior inference under the finite-infinite shared atoms nested (fiSAN) model with Gaussian likelihood. The model uses a Dirichlet process mixture prior at the distributional level, and Dirichlet mixture with an unknown number of components (following the specification of Frühwirth-Schnatter et al., 2021) at the observational level. The algorithm for the nonparametric component is based on the slice sampler for DPM of Kalli, Griffin and Walker (2011).

Usage

```
sample_fiSAN(nrep, burn, y, group,
            maxK = 50, maxL = 50,
            m0 = 0, tau0 = 0.1, lambda0 = 3, gamma0 = 2,
            hyp_alpha1 = 1, hyp_alpha2 = 1,
            hyp_beta1 = 6, hyp_beta2 = 3,
            eps_beta = NULL,
            alpha = NULL, beta = NULL,
            warmstart = TRUE, nclus_start = NULL,
            mu_start = NULL, sigma2_start = NULL,
            M_start = NULL, S_start = NULL,
            alpha_start = NULL, beta_start = NULL,
            progress = TRUE, seed = NULL)
```

Arguments

`nrep`	Number of MCMC iterations.
`burn`	Number of discarded iterations.
`y`	Vector of observations.
`group`	Vector of the same length of y indicating the group membership (numeric).
`maxK`	Maximum number of distributional clusters K (default = 50).
`maxL`	Maximum number of observational clusters L (default = 50).
`m0, tau0, lambda0, gamma0`	Hyperparameters on $(\mu, \sigma^2) \sim NIG(m_0, \tau_0, \lambda_0, \gamma_0)$. Default is (0, 0.1, 3, 2).
`hyp_alpha1, hyp_alpha2`	If a random α is used, (`hyp_alpha1`,`hyp_alpha2`) specify the hyperparameters (default = (1,1)). The prior is α ~ Gamma(`hyp_alpha1`, `hyp_alpha2`).
`hyp_beta1, hyp_beta2, eps_beta`	If a random β is used, (`hyp_beta1`,`hyp_beta2`) specifies the hyperparameter (default = (6,3)). The prior is $\beta \sim Gamma($`hyp_beta1`, `hyp_beta2`$)$. In this case, `eps_beta` is the tuning parameter of the MH step.
`alpha`	Distributional DP parameter if fixed (optional). The distribution is $\pi \sim GEM(\alpha)$.

beta	Observational Dirichlet parameter if fixed (optional). The distribution is Dirichlet(`rep(beta/maxL, maxL)`).
warmstart, nclus_start	
	Initialization of the observational clustering. `warmstart` is logical parameter (default = `TRUE`) of whether a kmeans clustering should be used to initialize the chains. An initial guess of the number of observational clusters can be passed via the `nclus_start` parameter (optional)
mu_start, sigma2_start, M_start, S_start, alpha_start, beta_start	
	Starting points of the MCMC chains (optional). Default is `nclus_start = min(c(maxL, 30))`. `mu_start`, `sigma2_start` are vectors of length `maxL`. `M_start` is a vector of observational cluster allocation of length N. `S_start` is a vector of observational cluster allocation of length J. `alpha_start`, `alpha_start` are numeric.
progress	show a progress bar? (logical, default TRUE).
seed	set a fixed seed.

Details

Data structure

The finite-infinite common atoms mixture model is used to perform inference in nested settings, where the data are organized into J groups. The data should be continuous observations $(Y_1, \ldots, Y_J)$, where each $Y_j = (y_{1,j}, \ldots, y_{n_j,j})$ contains the n_j observations from group j, for $j = 1, \ldots, J$. The function takes as input the data as a numeric vector y in this concatenated form. Hence y should be a vector of length $n_1 + \cdots + n_J$. The group parameter is a numeric vector of the same size as y indicating the group membership for each individual observation. Notice that with this specification the observations in the same group need not be contiguous as long as the correspondence between the variables y and group is maintained.

Model

The data are modeled using a univariate Gaussian likelihood, where both the mean and the variance are observational-cluster-specific, i.e.,

$$y_{i,j} \mid M_{i,j} = l \sim N(\mu_l, \sigma_l^2)$$

where $M_{i,j} \in \{1, \ldots, L\}$ is the observational cluster indicator of observation i in group j. The prior on the model parameters is a Normal-Inverse-Gamma distribution $(\mu_l, \sigma_l^2) \sim NIG(m_0, \tau_0, \lambda_0, \gamma_0)$, i.e., $\mu_l \mid \sigma_l^2 \sim N(m_0, \sigma_l^2/\tau_0)$, $1/\sigma_l^2 \sim Gamma(\lambda_0, \gamma_0)$ (shape, rate).

Clustering

The model performs a clustering of both observations and groups. The clustering of groups (distributional clustering) is provided by the allocation variables $S_j \in \{1, 2, \ldots\}$, with

$$Pr(S_j = k \mid \ldots) = \pi_k \qquad \text{for } k = 1, 2, \ldots$$

The distribution of the probabilities is $\{\pi_k\}_{k=1}^{\infty} \sim GEM(\alpha)$, where GEM is the Griffiths-Engen-McCloskey distribution of parameter α, which characterizes the stick-breaking construction of the DP (Sethuraman, 1994).

The clustering of observations (observational clustering) is provided by the allocation variables $M_{i,j} \in \{1, \ldots, L\}$, with

$$Pr(M_{i,j} = l \mid S_j = k, \ldots) = \omega_{l,k} \qquad \text{for } k = 1, 2, \ldots; l = 1, \ldots, L.$$

The distribution of the probabilities is $(\omega_{1,k}, \ldots, \omega_{L,k}) \sim Dirichlet_L(\beta/L, \ldots, \beta/L)$ for all $k = 1, 2, \ldots$. Moreover, the dimension L is random (see Frühwirth-Schnatter et al., 2021).

Value

sample_fiSAN returns four objects:

- model: name of the fitted model.
- params: list containing the data and the parameters used in the simulation. Details below.
- sim: list containing the simulated values (MCMC chains). Details below.
- time: total computation time.

Data and parameters: params is a list with the following components:

nrep Number of MCMC iterations.

y, group Data and group vectors.

maxK, maxL Maximum number of distributional and observational clusters.

m0, tau0, lambda0, gamma0 Model hyperparameters.

(hyp_alpha1,hyp_alpha2) **or** alpha Either the hyperparameters on α (if α random), or the value for α (if fixed).

(hyp_beta1, hyp_beta2, eps_beta) **or** beta Either the hyperparameters on β and MH step size (if β random), or the value for β (if fixed).

Simulated values: sim is a list with the following components:

mu Matrix of size (nrep, maxL). Each row is a posterior sample of the mean parameter for each observational cluster $(\mu_1, \ldots \mu_L)$.

sigma2 Matrix of size (nrep, maxL). Each row is a posterior sample of the variance parameter for each observational cluster $(\sigma_1^2, \ldots \sigma_L^2)$.

obs_cluster Matrix of size (nrep, n), with n = length(y). Each row is a posterior sample of the observational cluster allocation variables $(M_{1,1}, \ldots, M_{n_J,J})$.

distr_cluster Matrix of size (nrep, J), with J = length(unique(group)). Each row is a posterior sample of the distributional cluster allocation variables $(S_1, \ldots, S_J)$.

pi Matrix of size (nrep, maxK). Each row is a posterior sample of the distributional cluster probabilities $(\pi_1, \ldots, \pi_{maxK})$.

omega 3-d array of size (maxL, maxK, nrep). Each slice is a posterior sample of the observational cluster probabilities. In each slice, each column k is a vector (of length maxL) observational cluster probabilities $(\omega_{1,k}, \ldots, \omega_{L,k})$ for distributional cluster k.

alpha Vector of length nrep of posterior samples of the parameter α.

beta Vector of length nrep of posterior samples of the parameter β.

maxK Vector of length nrep of the number of distributional DP components used by the slice sampler.

L_plus Vector of length nrep of posterior samples of the number of observational clusters.

L Vector of length nrep of posterior samples of the observational Dirichlet dimension.

References

Frühwirth-Schnatter, S., Malsiner-Walli, G. and Grün, B. (2021). Generalized mixtures of finite mixtures and telescoping sampling. *Bayesian Analysis*, 16(4), 1279–1307. <doi:10.1214/21-BA1294>

Kalli, M., Griffin, J.E., and Walker, S.G. (2011). Slice Sampling Mixture Models, *Statistics and Computing*, 21, 93–105. <doi:10.1007/s11222-009-9150-y>

Sethuraman, A.J. (1994). A Constructive Definition of Dirichlet Priors, *Statistica Sinica*, 4, 639–650.

Examples

```
set.seed(123)
y <- c(rnorm(40,0,0.3), rnorm(20,5,0.3))
g <- c(rep(1,30), rep(2, 30))
plot(density(y[g==1]), xlim = c(-5,10))
lines(density(y[g==2]), col = 2)
out <- sample_fiSAN(nrep = 500, burn = 200, y = y, group = g,
                    nclus_start = 2,
                    maxK = 20, maxL = 20,
                    beta = 1)
out
```

---

sample_fiSAN_sparsemix

Sample fiSAN with sparse mixtures

Description

`sample_fiSAN_sparsemix` is used to perform posterior inference under the finite-infinite shared atoms nested (fiSAN) model with Gaussian likelihood. The model uses a Dirichlet process mixture prior at the distributional level, and a sparse (overfitted) Dirichlet mixture (Malsiner-Walli et al., 2016) at the observational level. The algorithm for the nonparametric component is based on the slice sampler for DPM of Kalli, Griffin and Walker (2011).

Usage

```
sample_fiSAN_sparsemix(nrep, burn, y, group,
                maxK = 50, maxL = 50,
                m0 = 0, tau0 = 0.1, lambda0 = 3, gamma0 = 2,
                hyp_alpha1 = 1, hyp_alpha2 = 1,
                hyp_beta = 10,
                eps_beta = NULL,
                alpha = NULL, beta = NULL,
                warmstart = TRUE, nclus_start = NULL,
                mu_start = NULL, sigma2_start = NULL,
                M_start = NULL, S_start = NULL,
                alpha_start = NULL, beta_start = NULL,
                progress = TRUE, seed = NULL)
```

Arguments

`nrep`	Number of MCMC iterations.
`burn`	Number of discarded iterations.
`y`	Vector of observations.
`group`	Vector of the same length of y indicating the group membership (numeric).
`maxK`	Maximum number of distributional clusters K (default = 50).
`maxL`	Maximum number of observational clusters L (default = 50).

`m0, tau0, lambda0, gamma0`
> Hyperparameters on $(\mu, \sigma^2) \sim NIG(m_0, \tau_0, \lambda_0, \gamma_0)$. Default is (0, 0.1, 3, 2).

`hyp_alpha1, hyp_alpha2`
> If a random α is used, (`hyp_alpha1`,`hyp_alpha2`) specify the hyperparameters (default = (1,1)). The prior is α ~ Gamma(hyp_alpha1, hyp_alpha2).

`hyp_beta, eps_beta`
> If a random β is used, `hyp_beta` specifies the hyperparameter (default = 10). The prior is β ~ Gamma(hyp_beta, hyp_beta*maxL), following Malsiner-Walli et al. (2016). In this case, `eps_beta` is the tuning parameter of the MH step.

`alpha`	Distributional DP parameter if fixed (optional). The distribution is $\pi \sim GEM(\alpha)$.
`beta`	Observational Dirichlet parameter if fixed (optional). The distribution is Dirichlet(`rep(beta, maxL)`). Notice that beta should be small to ensure sparsity (e.g. beta = 0.01)

`warmstart, nclus_start`
> Initialization of the observational clustering. `warmstart` is logical parameter (default = TRUE) of whether a kmeans clustering should be used to initialize the chains. An initial guess of the number of observational clusters can be passed via the `nclus_start` parameter (optional)

`mu_start, sigma2_start, M_start, S_start, alpha_start, beta_start`
> Starting points of the MCMC chains (optional). Default is nclus_start = min(c(maxL, 30)). `mu_start`, `sigma2_start` are vectors of length maxL. `M_start` is a vector of observational cluster allocation of length N. `S_start` is a vector of observational cluster allocation of length J. `alpha_start`, `alpha_start` are numeric.

`progress`	show a progress bar? (logical, default TRUE).
`seed`	set a fixed seed.

Details

Data structure

The finite-infinite common atoms mixture model is used to perform inference in nested settings, where the data are organized into J groups. The data should be continuous observations $(Y_1, \ldots, Y_J)$, where each $Y_j = (y_{1,j}, \ldots, y_{n_j,j})$ contains the n_j observations from group j, for $j = 1, \ldots, J$. The function takes as input the data as a numeric vector y in this concatenated form. Hence y should be a vector of length $n_1 + \cdots + n_J$. The `group` parameter is a numeric vector of the same size as y indicating the group membership for each individual observation. Notice that with this specification

the observations in the same group need not be contiguous as long as the correspondence between the variables y and group is maintained.

Model

The data are modeled using a univariate Gaussian likelihood, where both the mean and the variance are observational-cluster-specific, i.e.,

$$y_{i,j} \mid M_{i,j} = l \sim N(\mu_l, \sigma_l^2)$$

where $M_{i,j} \in \{1, \dots, L\}$ is the observational cluster indicator of observation i in group j. The prior on the model parameters is a Normal-Inverse-Gamma distribution $(\mu_l, \sigma_l^2) \sim NIG(m_0, \tau_0, \lambda_0, \gamma_0)$, i.e., $\mu_l \mid \sigma_l^2 \sim N(m_0, \sigma_l^2/\tau_0)$, $1/\sigma_l^2 \sim Gamma(\lambda_0, \gamma_0)$ (shape, rate).

Clustering

The model performs a clustering of both observations and groups. The clustering of groups (distributional clustering) is provided by the allocation variables $S_j \in \{1, 2, \dots\}$, with

$$Pr(S_j = k \mid \dots) = \pi_k \qquad \text{for } k = 1, 2, \dots$$

The distribution of the probabilities is $\{\pi_k\}_{k=1}^{\infty} \sim GEM(\alpha)$, where GEM is the Griffiths-Engen-McCloskey distribution of parameter α, which characterizes the stick-breaking construction of the DP (Sethuraman, 1994).

The clustering of observations (observational clustering) is provided by the allocation variables $M_{i,j} \in \{1, \dots, L\}$, with

$$Pr(M_{i,j} = l \mid S_j = k, \dots) = \omega_{l,k} \qquad \text{for } k = 1, 2, \dots; \ l = 1, \dots, L.$$

The distribution of the probabilities is $(\omega_{1,k}, \dots, \omega_{L,k}) \sim Dirichlet_L(\beta, \dots, \beta)$ for all $k = 1, 2, \dots$.

Value

sample_fiSAN_sparsemix returns four objects:

- model: name of the fitted model.
- params: list containing the data and the parameters used in the simulation. Details below.
- sim: list containing the simulated values (MCMC chains). Details below.
- time: total computation time.

Data and parameters: params is a list with the following components:

nrep Number of MCMC iterations.

y, group Data and group vectors.

maxK, maxL Maximum number of distributional and observational clusters.

m0, tau0, lambda0, gamma0 Model hyperparameters.

(hyp_alpha1,hyp_alpha2) **or** alpha Either the hyperparameters on α (if α random), or the value for α (if fixed).

(hyp_beta,eps_beta) **or** beta Either the hyperparameter on β and MH step size (if β random), or the value for β (if fixed).

Simulated values: `sim` is a list with the following components:

mu Matrix of size (`nrep`, `maxL`). Each row is a posterior sample of the mean parameter for each observational cluster $(\mu_1, \ldots \mu_L)$.

sigma2 Matrix of size (`nrep`, `maxL`). Each row is a posterior sample of the variance parameter for each observational cluster $(\sigma_1^2, \ldots \sigma_L^2)$.

obs_cluster Matrix of size (`nrep`, n), with n = `length(y)`. Each row is a posterior sample of the observational cluster allocation variables $(M_{1,1}, \ldots, M_{n_J,J})$.

distr_cluster Matrix of size (`nrep`, J), with J = `length(unique(group))`. Each row is a posterior sample of the distributional cluster allocation variables $(S_1, \ldots, S_J)$.

pi Matrix of size (`nrep`, `maxK`). Each row is a posterior sample of the distributional cluster probabilities $(\pi_1, \ldots, \pi_{maxK})$.

omega 3-d array of size (`maxL`, `maxK`, `nrep`). Each slice is a posterior sample of the observational cluster probabilities. In each slice, each column k is a vector (of length `maxL`) observational cluster probabilities $(\omega_{1,k}, \ldots, \omega_{L,k})$ for distributional cluster k.

alpha Vector of length `nrep` of posterior samples of the parameter α.

beta Vector of length `nrep` of posterior samples of the parameter β.

maxK Vector of length `nrep` of the number of distributional DP components used by the slice sampler.

References

Kalli, M., Griffin, J.E., and Walker, S.G. (2011). Slice Sampling Mixture Models, *Statistics and Computing*, 21, 93–105. <doi:10.1007/s11222-009-9150-y>

Malsiner-Walli, G., Frühwirth-Schnatter, S. and Grün, B. (2016). Model-based clustering based on sparse finite Gaussian mixtures. Statistics and Computing 26, 303–324. <doi:10.1007/s11222-014-9500-2>

Sethuraman, A.J. (1994). A Constructive Definition of Dirichlet Priors, *Statistica Sinica*, 4, 639–650.

Examples

```
set.seed(123)
y <- c(rnorm(40,0,0.3), rnorm(20,5,0.3))
g <- c(rep(1,30), rep(2, 30))
plot(density(y[g==1]), xlim = c(-5,10))
lines(density(y[g==2]), col = 2)
out <- sample_fiSAN_sparsemix(nrep = 500, burn = 200, y = y, group = g,
                              nclus_start = 2,
                              maxK = 20, maxL = 20,
                              beta = 0.01)
out
```

---

sample_fSAN   *Sample fSAN*

Description

`sample_fSAN` is used to perform posterior inference under the finite shared atoms nested (fSAN) model with Gaussian likelihood (originally proposed in D'Angelo et al., 2023). The model uses Dirichlet mixtures with an unknown number of components (following the specification of Frühwirth-Schnatter et al., 2021) at both the observational and distributional level.

Usage

```
sample_fSAN(nrep, burn, y, group,
            maxK = 50, maxL = 50,
            m0 = 0, tau0 = 0.1, lambda0 = 3, gamma0 = 2,
            hyp_alpha1 = 6, hyp_alpha2 = 3,
            hyp_beta1 = 6, hyp_beta2 = 3,
            eps_alpha = NULL, eps_beta = NULL,
            alpha = NULL, beta = NULL,
            warmstart = TRUE, nclus_start = NULL,
            mu_start = NULL, sigma2_start = NULL,
            M_start = NULL, S_start = NULL,
            alpha_start = NULL, beta_start = NULL,
            progress = TRUE, seed = NULL)
```

Arguments

nrep	Number of MCMC iterations.
burn	Number of discarded iterations.
y	Vector of observations.
group	Vector of the same length of y indicating the group membership (numeric).
maxK	Maximum number of distributional clusters K (default = 50).
maxL	Maximum number of observational clusters L (default = 50).
m0, tau0, lambda0, gamma0	
	Hyperparameters on $(\mu, \sigma^2) \sim NIG(m_0, \tau_0, \lambda_0, \gamma_0)$. Default is (0, 0.1, 3, 2).
hyp_alpha1, hyp_alpha2, eps_alpha	
	If a random α is used, (`hyp_alpha1`,`hyp_alpha2`) specify the hyperparameters (default = (6,3)). The prior is $\alpha \sim Gamma$(`hyp_alpha1`, `hyp_alpha2`). In this case, `eps_alpha` is the tuning parameter of the MH step.
hyp_beta1, hyp_beta2, eps_beta	
	If a random β is used, (`hyp_beta1`,`hyp_beta2`) specifies the hyperparameter (default = (6,3)). The prior is $\beta \sim Gamma$(`hyp_beta1`, `hyp_beta2`). In this case, `eps_beta` is the tuning parameter of the MH step.
alpha	Distributional Dirichlet parameter if fixed (optional). The distribution is Dirichlet(`rep(alpha/maxK, maxK)`).

beta	Observational Dirichlet parameter if fixed (optional). The distribution is Dirichlet(`rep(beta/maxL, maxL)`).
warmstart, nclus_start	
	Initialization of the observational clustering. `warmstart` is logical parameter (default = TRUE) of whether a kmeans clustering should be used to initialize the chains. An initial guess of the number of observational clusters can be passed via the `nclus_start` parameter (optional). Default is `nclus_start = min(c(maxL, 30))`.
mu_start, sigma2_start, M_start, S_start, alpha_start, beta_start	
	Starting points of the MCMC chains (optional). `mu_start, sigma2_start` are vectors of length `maxL`. `M_start` is a vector of observational cluster allocation of length N. `S_start` is a vector of observational cluster allocation of length J. `alpha_start, alpha_start` are numeric.
progress	show a progress bar? (logical, default TRUE.)
seed	set a fixed seed.

Details

Data structure

The finite common atoms mixture model is used to perform inference in nested settings, where the data are organized into J groups. The data should be continuous observations $(Y_1, \ldots, Y_J)$, where each $Y_j = (y_{1,j}, \ldots, y_{n_j,j})$ contains the n_j observations from group j, for $j = 1, \ldots, J$. The function takes as input the data as a numeric vector y in this concatenated form. Hence y should be a vector of length $n_1 + \cdots + n_J$. The group parameter is a numeric vector of the same size as y indicating the group membership for each individual observation. Notice that with this specification the observations in the same group need not be contiguous as long as the correspondence between the variables y and group is maintained.

Model

The data are modeled using a univariate Gaussian likelihood, where both the mean and the variance are observational-cluster-specific, i.e.,

$$y_{i,j} \mid M_{i,j} = l \sim N(\mu_l, \sigma_l^2)$$

where $M_{i,j} \in \{1, \ldots, L\}$ is the observational cluster indicator of observation i in group j. The prior on the model parameters is a Normal-Inverse-Gamma distribution $(\mu_l, \sigma_l^2) \sim NIG(m_0, \tau_0, \lambda_0, \gamma_0)$, i.e., $\mu_l \mid \sigma_l^2 \sim N(m_0, \sigma_l^2/\tau_0)$, $1/\sigma_l^2 \sim Gamma(\lambda_0, \gamma_0)$ (shape, rate).

Clustering

The model performs a clustering of both observations and groups. The clustering of groups (distributional clustering) is provided by the allocation variables $S_j \in \{1, \ldots, K\}$, with

$$Pr(S_j = k \mid \ldots) = \pi_k \qquad \text{for } k = 1, \ldots, K.$$

The distribution of the probabilities is $(\pi_1, \ldots, \pi_K) \sim Dirichlet_K(\alpha/K, \ldots, \alpha/K)$. Moreover, the dimension K is random (see Frühwirth-Schnatter et al., 2021).

The clustering of observations (observational clustering) is provided by the allocation variables $M_{i,j} \in \{1, \ldots, L\}$, with

$$Pr(M_{i,j} = l \mid S_j = k, \ldots) = \omega_{l,k} \qquad \text{for } k = 1, \ldots, K; l = 1, \ldots, L.$$

The distribution of the probabilities is $(\omega_{1,k}, \ldots, \omega_{L,k}) \sim Dirichlet_L(\beta/L, \ldots, \beta/L)$ for all $k = 1, \ldots, K$. Moreover, the dimension L is random (see Frühwirth-Schnatter et al., 2021).

Value

sample_fSAN returns four objects:

- model: name of the fitted model.
- params: list containing the data and the parameters used in the simulation. Details below.
- sim: list containing the simulated values (MCMC chains). Details below.
- time: total computation time.

Data and parameters: params is a list with the following components:

nrep Number of MCMC iterations.

y, group Data and group vectors.

maxK, maxL Maximum number of distributional and observational clusters.

m0, tau0, lambda0, gamma0 Model hyperparameters.

(hyp_alpha1, hyp_alpha2, eps_alpha) **or** alpha Either the hyperparameters on α and MH step size (if α random), or the value for α (if fixed).

(hyp_beta1, hyp_beta2, eps_beta) **or** beta Either the hyperparameters on β and MH step size (if β random), or the value for β (if fixed).

Simulated values: sim is a list with the following components:

mu Matrix of size (nrep, maxL). Each row is a posterior sample of the mean parameter for each observational cluster $(\mu_1, \ldots \mu_L)$.

sigma2 Matrix of size (nrep, maxL). Each row is a posterior sample of the variance parameter for each observational cluster $(\sigma_1^2, \ldots \sigma_L^2)$.

obs_cluster Matrix of size (nrep, n), with n = length(y). Each row is a posterior sample of the observational cluster allocation variables $(M_{1,1}, \ldots, M_{n_J,J})$.

distr_cluster Matrix of size (nrep, J), with J = length(unique(group)). Each row is a posterior sample of the distributional cluster allocation variables $(S_1, \ldots, S_J)$.

pi Matrix of size (nrep, maxK). Each row is a posterior sample of the distributional cluster probabilities $(\pi_1, \ldots, \pi_K)$.

omega 3-d array of size (maxL, maxK, nrep). Each slice is a posterior sample of the observational cluster probabilities. In each slice, each column k is a vector (of length maxL) observational cluster probabilities $(\omega_{1,k}, \ldots, \omega_{L,k})$ for distributional cluster k.

alpha Vector of length nrep of posterior samples of the parameter α.

beta Vector of length nrep of posterior samples of the parameter β.

K_plus Vector of length nrep of posterior samples of the number of distributional clusters.

L_plus Vector of length nrep of posterior samples of the number of observational clusters.

K Vector of length nrep of posterior samples of the distributional Dirichlet dimension.

L Vector of length nrep of posterior samples of the observational Dirichlet dimension.

References

D'Angelo, L., Canale, A., Yu, Z., and Guindani, M. (2023). Bayesian nonparametric analysis for the detection of spikes in noisy calcium imaging data. *Biometrics*, 79(2), 1370–1382. <doi:10.1111/biom.13626>

Frühwirth-Schnatter, S., Malsiner-Walli, G. and Grün, B. (2021). Generalized mixtures of finite mixtures and telescoping sampling. *Bayesian Analysis*, 16(4), 1279–1307. <doi:10.1214/21-BA1294>

Examples

```
set.seed(123)
y <- c(rnorm(40,0,0.3), rnorm(20,5,0.3))
g <- c(rep(1,30), rep(2, 30))
plot(density(y[g==1]), xlim = c(-5,10))
lines(density(y[g==2]), col = 2)
out <- sample_fSAN(nrep = 500, burn = 200,
                   y = y, group = g,
                   nclus_start = 2,
                   maxK = 20, maxL = 20,
                   alpha = 1, beta = 1)
out
```

---

sample_fSAN_sparsemix   *Sample fSAN with sparse mixtures*

Description

sample_fSAN_sparsemix is used to perform posterior inference under the finite shared atoms nested (fSAN) model with Gaussian likelihood (originally proposed in D'Angelo et al., 2023). The model uses overfitted (sparse) Dirichlet mixtures (Malsiner-Walli et al., 2016) at both the observational and distributional level.

Usage

```
sample_fSAN_sparsemix(nrep, burn, y, group,
             maxK = 50, maxL = 50,
             m0 = 0, tau0 = 0.1, lambda0 = 3, gamma0 = 2,
             hyp_alpha = 10, hyp_beta = 10,
             eps_alpha = NULL, eps_beta = NULL,
             alpha = NULL, beta = NULL,
             warmstart = TRUE, nclus_start = NULL,
             mu_start = NULL, sigma2_start = NULL,
             M_start = NULL, S_start = NULL,
             alpha_start = NULL, beta_start = NULL,
             progress = TRUE, seed = NULL)
```

Arguments

`nrep`	Number of MCMC iterations.
`burn`	Number of discarded iterations.
`y`	Vector of observations.
`group`	Vector of the same length of y indicating the group membership (numeric).
`maxK`	Maximum number of distributional clusters K (default = 50).
`maxL`	Maximum number of observational clusters L (default = 50).

`m0, tau0, lambda0, gamma0`
Hyperparameters on $(\mu, \sigma^2) \sim NIG(m_0, \tau_0, \lambda_0, \gamma_0)$. Default is $(0, 0.1, 3, 2)$.

`hyp_alpha, eps_alpha`
If a random α is used, `hyp_alpha` specifies the hyperparameter (default = 10). The prior is $\alpha \sim$ Gamma(`hyp_alpha`, `hyp_alpha*maxK`), following Malsiner-Walli et al. (2016). In this case, `eps_alpha` is the tuning parameter of the MH step.

`hyp_beta, eps_beta`
If a random β is used, `hyp_beta` specifies the hyperparameter (default = 10). The prior is $\beta \sim$ Gamma(`hyp_beta`, `hyp_beta*maxL`), following Malsiner-Walli et al. (2016). In this case, `eps_beta` is the tuning parameter of the MH step.

`alpha`	Distributional Dirichlet parameter if fixed (optional). The distribution is Dirichlet(`rep(alpha, maxK)`).
`beta`	Observational Dirichlet parameter if fixed (optional). The distribution is Dirichlet(`rep(beta, maxL)`).

`warmstart, nclus_start`
Initialization of the observational clustering. `warmstart` is logical parameter (default = `TRUE`) of whether a kmeans clustering should be used to initialize the chains. An initial guess of the number of observational clusters can be passed via the `nclus_start` parameter (optional)

`mu_start, sigma2_start, M_start, S_start, alpha_start, beta_start`
Starting points of the MCMC chains (optional). Default is `nclus_start = min(c(maxL, 30))`. `mu_start, sigma2_start` are vectors of length `maxL`. `M_start` is a vector of observational cluster allocation of length N. `S_start` is a vector of observational cluster allocation of length J. `alpha_start, alpha_start` are numeric.

`progress`	show a progress bar? (logical, default TRUE).
`seed`	set a fixed seed.

Details

Data structure

The overfitted mixture common atoms model is used to perform inference in nested settings, where the data are organized into J groups. The data should be continuous observations $(Y_1, \ldots, Y_J)$, where each $Y_j = (y_{1,j}, \ldots, y_{n_j,j})$ contains the n_j observations from group j, for $j = 1, \ldots, J$. The function takes as input the data as a numeric vector y in this concatenated form. Hence y should be a vector of length $n_1 + \cdots + n_J$. The group parameter is a numeric vector of the same size as y

indicating the group membership for each individual observation. Notice that with this specification the observations in the same group need not be contiguous as long as the correspondence between the variables y and group is maintained.

Model

The data are modeled using a univariate Gaussian likelihood, where both the mean and the variance are observational-cluster-specific, i.e.,

$$y_{i,j} \mid M_{i,j} = l \sim N(\mu_l, \sigma_l^2)$$

where $M_{i,j} \in \{1, \ldots, L\}$ is the observational cluster indicator of observation i in group j. The prior on the model parameters is a Normal-Inverse-Gamma distribution $(\mu_l, \sigma_l^2) \sim NIG(m_0, \tau_0, \lambda_0, \gamma_0)$, i.e., $\mu_l \mid \sigma_l^2 \sim N(m_0, \sigma_l^2/\tau_0)$, $1/\sigma_l^2 \sim Gamma(\lambda_0, \gamma_0)$ (shape, rate).

Clustering

The model performs a clustering of both observations and groups. The clustering of groups (distributional clustering) is provided by the allocation variables $S_j \in \{1, \ldots, K\}$, with

$$Pr(S_j = k \mid \ldots) = \pi_k \qquad \text{for } k = 1, \ldots, K.$$

The distribution of the probabilities is $(\pi_1, \ldots, \pi_K) \sim Dirichlet_K(\alpha, \ldots, \alpha)$.

The clustering of observations (observational clustering) is provided by the allocation variables $M_{i,j} \in \{1, \ldots, L\}$, with

$$Pr(M_{i,j} = l \mid S_j = k, \ldots) = \omega_{l,k} \qquad \text{for } k = 1, \ldots, K; \ l = 1, \ldots, L.$$

The distribution of the probabilities is $(\omega_{1,k}, \ldots, \omega_{L,k}) \sim Dirichlet_L(\beta, \ldots, \beta)$ for all $k = 1, \ldots, K$.

Value

sample_fSAN_sparsemix returns four objects:

- model: name of the fitted model.
- params: list containing the data and the parameters used in the simulation. Details below.
- sim: list containing the simulated values (MCMC chains). Details below.
- time: total computation time.

Data and parameters: params is a list with the following components:

nrep Number of MCMC iterations.

y, group Data and group vectors.

maxK, maxL Maximum number of distributional and observational clusters.

m0, tau0, lambda0, gamma0 Model hyperparameters.

(hyp_alpha, eps_alpha) **or** alpha Either the hyperparameter on α and MH step size (if α random), or the value for α (if fixed).

(hyp_beta, eps_beta) **or** beta Either the hyperparameter on β and MH step size (if β random), or the value for β (if fixed).

Simulated values: `sim` is a list with the following components:

mu Matrix of size (`nrep`, `maxL`). Each row is a posterior sample of the mean parameter for each observational cluster $(\mu_1, \ldots \mu_L)$.

sigma2 Matrix of size (`nrep`, `maxL`). Each row is a posterior sample of the variance parameter for each observational cluster $(\sigma_1^2, \ldots \sigma_L^2)$.

obs_cluster Matrix of size (`nrep`, n), with n = `length(y)`. Each row is a posterior sample of the observational cluster allocation variables $(M_{1,1}, \ldots, M_{n_J,J})$.

distr_cluster Matrix of size (`nrep`, J), with J = `length(unique(group))`. Each row is a posterior sample of the distributional cluster allocation variables $(S_1, \ldots, S_J)$.

pi Matrix of size (`nrep`, `maxK`). Each row is a posterior sample of the distributional cluster probabilities $(\pi_1, \ldots, \pi_K)$.

omega 3-d array of size (`maxL`, `maxK`, `nrep`). Each slice is a posterior sample of the observational cluster probabilities. In each slice, each column k is a vector (of length `maxL`) observational cluster probabilities $(\omega_{1,k}, \ldots, \omega_{L,k})$ for distributional cluster k.

alpha Vector of length `nrep` of posterior samples of the parameter α.

beta Vector of length `nrep` of posterior samples of the parameter β.

References

D'Angelo, L., Canale, A., Yu, Z., and Guindani, M. (2023). Bayesian nonparametric analysis for the detection of spikes in noisy calcium imaging data. *Biometrics*, 79(2), 1370–1382. <doi:10.1111/biom.13626>

Malsiner-Walli, G., Frühwirth-Schnatter, S. and Grün, B. (2016). Model-based clustering based on sparse finite Gaussian mixtures. Statistics and Computing 26, 303–324. <doi:10.1007/s11222-014-9500-2>

Examples

```
set.seed(123)
y <- c(rnorm(40,0,0.3), rnorm(20,5,0.3))
g <- c(rep(1,30), rep(2, 30))
plot(density(y[g==1]), xlim = c(-5,10))
lines(density(y[g==2]), col = 2)
out <- sample_fSAN_sparsemix(nrep = 500, burn = 200, y = y, group = g,
                             nclus_start = 2,
                             maxK = 20, maxL = 20,
                             alpha = 0.01, beta = 0.01)
out
```

---

traceplot   *Traceplot: plot MCMC chains*

Description

Check the convergence of the MCMC through visual inspection of the chains.

Usage

```
traceplot(object, params,
          show_density = TRUE,
          show_burnin = TRUE,
          length_burnin = NULL,
          show_convergence = TRUE,
          trunc_plot = 10)
```

Arguments

object	object of class SANmcmc (the result of a call to `sample_fiSAN`, `sample_fiSAN_sparsemix`, `sample_fSAN`, `sample_fSAN_sparsemix`, or `sample_CAM`).
params	vector of strings with the names of the parameters to check.
show_density	logical (default TRUE). Whether a kernel estimate of the density should be plotted. The burn-in is always discarded.
show_burnin	logical (default TRUE). Whether the first part of the chains should be plotted in the traceplots.
length_burnin	if show_burnin = FALSE, the length of the burn-in to be discarded.
show_convergence	
	logical (default TRUE). Whether a superimposed red line of the cumulative mean should be plotted.
trunc_plot	integer (default = 10). For multidimensional parameters, the maximum number of components to be plotted.

Value

The function displays the traceplots of the MCMC algorithm.

Note

The function is not available for the observational weights ω.

Examples

```
set.seed(123)
y <- c(rnorm(40,0,0.3), rnorm(20,5,0.3))
g <- c(rep(1,30), rep(2, 30))
out <- sample_fiSAN(nrep = 500, burn = 200,
```

```
                         y = y, group = g,
                         nclus_start = 2,
                         maxK = 20, maxL = 20,
                         beta = 1)
traceplot(out, params = c("mu", "sigma2"), trunc_plot = 2)
```

Index