Package ‘SANvi’

May 15, 2024

Type    Package
Title   Fitting Shared Atoms Nested Models via Variational Bayes
Version 0.1.1
Date    2024-05-14
Maintainer Francesco Denti <francescodenti.personal@gmail.com>

URL  https://github.com/fradenti/SANvi

BugReports https://github.com/fradenti/SANvi/issues

Description An efficient tool for fitting the nested common and shared atoms models using variational Bayes approximate inference for fast computation. 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 analyze the results of the fitting procedure.


License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.1

Imports Rcpp, matrixStats

Depends scales, RColorBrewer

LinkingTo Rcpp, RcppArmadillo

Language en-US

NeedsCompilation yes

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

Repository CRAN

Date/Publication 2024-05-15 11:40:06 UTC
estimate_atoms_weights_vi

Estimate the Posterior Atoms and Weights of the Discrete Mixing Distributions

Description

This function estimates the posterior atoms and weights characterizing the discrete mixing distributions using the variational estimates obtained from one of the model implemented in SANvi.

Usage

```r
estimate_atoms_weights_vi(output)
```

```r
## S3 method for class 'vi_atoms_weights'
plot(x, DC_num = NULL, lim = 2, ...)
```

```r
## S3 method for class 'vi_atoms_weights'
print(x, thr = 0.01, ...)
```

Arguments

- `output`: an object of class `SANvb`, which is the output of one of the variational functions `variational_CAM`, `variational_fiSAN`, `variational_fSAN`.
- `x`: an object of class `variational_estimates`, which can be obtained from the function `estimate_atoms_weights_vi`.
- `DC_num`: an integer or a vector of integers indicating which distributional clusters to plot.
- `lim`: optional value for `plot` method to adjust the limits of the x-axis. The atoms are plotted on a range given by `min(posterior means) - 2, max(posterior means) + 2`. Default is set to 2.
- `...`: ignored.
- `thr`: argument for the `print()` method. It should be a small positive number, representing a threshold. If the posterior weight of a shared atom is below the threshold, the atom is not reported.
Value

an object of class `vi_atoms_weights`, which is matrix comprising posterior means, variances, and a columns for each estimated DC containing the posterior weights.

See Also

variational_CAM, variational_fiSAN, variational_fSAN, extract_best.

Examples

```r
# Generate example data
set.seed(1232)
y <- c(rnorm(100), rnorm(100, 5))
g <- rep(1:2, rep(100, 2))

# Fitting fiSAN via variational inference
est <- variational_fiSAN(y, g, verbose = FALSE)

# Estimate posterior atoms and weights
estimate_atoms_weights_vi(est)
```

---

`estimate_clustering_vi`

*Estimate Posterior Clustering Assignments*

Description

This function estimates posterior clustering assignments based on posterior variational estimates obtained from one of the model implemented in SANvi.

Usage

`estimate_clustering_vi(output, ordered = TRUE)`

```r
## S3 method for class 'vi_clustering'
plot(
  x,
  DC_num = NULL,
  type = c("ecdf", "boxplot", "scatter"),
  palette_brewed = FALSE,
  ...
)

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

output an object of class SANvb, the output of one of the variational functions variational_CAM, variational_fiSAN, variational_fSAN.

ordered logical, if TRUE (default), the function sorts the distributional cluster labels reflecting the increasing values of medians of the data assigned to each DC.

x an object of class variational_estimates, which can be obtained from the function estimate_atoms_weights_vi.

DC_num an integer or a vector of integers indicating which distributional clusters to plot.

type what type of plot should be drawn (only for the left-side plot). Possible types are "boxplot", "ecdf", and "scatter".

palette_brewed (logical) the color palette to be used. Default is R base colors (palette_brewed = FALSE).

... ignored.

Value

a list of class clustering containing

• obs_level: a data frame containing the data values, their group indexes, the observational and distributional clustering assignments for each observation.

• dis_level: a vector with the distributional clustering assignment for each unit.

See Also

variational_CAM, variational_fiSAN, variational_fSAN, extract_best.

Examples

# Generate example data
set.seed(123)
y <- c(rnorm(100),rnorm(100,-5),rnorm(100,5),rnorm(100),
      rnorm(100),rnorm(100,-5),rnorm(100,5),rnorm(100))
g <- rep(1:4,rep(200,4))

# Fitting fiSAN via variational inference
est <- SANvi::variational_fiSAN(y,g,verbose = FALSE)

# Estimate clustering assignments
estimate_clustering_vi(est)
**extract_best**

Extract the best run from multiple trials

**Description**
A simple function to automatically extract the best run from a collection of fitted variational models.

**Usage**
```r
eextract_best(object)
```

**Arguments**
- **object**: an object of class `multistart`, obtained from the `variational_multistart` function.

**Value**
the single best run, an object of class SANvb.

**Examples**
```r
# Generate example dataset
set.seed(123)
y <- c(rnorm(100), rnorm(100, 5))
g <- rep(1:2, rep(100, 2))

# Estimate multiple models via variational inference
est <- variational_multistart(y, g, runs=5,
                              alpha_bar = 3, beta_bar = 3,
                              root=1234, warmstart = FALSE)

# Obtain best run
extract_best(est)
```

---

**plot.SANvb**

Plotting the variational inference output

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

## S3 method for class 'SANvb'
plot(x, ...)  

Arguments

x          object of class SANvb (the result of a call to \texttt{variational\_CAM}, \texttt{variational\_fiSAN}, \texttt{variational\_fSAN}.
...	      additional graphical parameters to be passed.

Value

The function plots a summary of the fitted model.

See Also

\texttt{print.SANvb}, \texttt{variational\_CAM}, \texttt{variational\_fiSAN}, \texttt{variational\_fSAN}.

\begin{verbatim}
\end{verbatim}

---

\texttt{print.SANvb} \hspace{1cm} \textit{Print variational inference output}

Description

Print method for objects of class SANvb.

Usage

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

Arguments

x          object of class SANvb (the result of a call to \texttt{variational\_CAM}, \texttt{variational\_fiSAN}, \texttt{variational\_fSAN}.
...	      further arguments passed to or from other methods.

Value

The function prints a summary of the fitted model.
variational_CAM

Mean Field Variational Bayes estimation of CAM

Description

variational_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.

Usage

```r
variational_CAM(y, group, maxL = 30, maxK = 20,
                  m0 = 0, tau0 = .01, lambda0 = 3, gamma0 = 2,
                  conc_hyperpar = c(1,1,1,1), conc_par = NULL,
                  epsilon = 1e-6, seed = NULL, maxSIM = 1e5,
                  warmstart = TRUE, verbose = FALSE)
```

Arguments

- `y` Numerical vector of observations (required).
- `group` Numerical vector of the same length of `y`, indicating the group membership (required).
- `maxL`, `maxK` integers, the upper bounds for the observational and distributional clusters to fit, respectively.
- `m0`, `tau0`, `lambda0`, `gamma0` Hyperparameters on $(\mu, \sigma^2) \sim NIG(m_0, \tau_0, \lambda_0, \gamma_0)$.
- `conc_hyperpar`, `conc_par` Vectors of values used for the concentration parameters of the stick-breaking representation for the distributional and observational DPs, respectively. The following two arguments can be passed. Specifically,
  - `conc_hyperpar` a vector with 4 positive entries: $(s_{\alpha_1}^a, s_{\alpha_2}^a, s_{\beta_1}^a, s_{\beta_2}^a)$. If a random concentration parameters $\alpha$ and $\beta$ are adopted, the specifications are $\alpha \sim Gamma(s_{\alpha_1}^a, s_{\alpha_2}^a)$ and $\beta \sim Gamma(s_{\beta_1}^a, s_{\beta_2}^a)$. Default set to unitary vector.
  - `conc_par` a vector with 2 positive entries: $(\alpha, \beta)$. Default is set to NULL. If specified, the previous argument is ignored and the two concentration parameters are assumed fixed and equal to $(\alpha, \beta)$.
- `epsilon` the tolerance that drives the convergence criterion adopted as stopping rule
- `seed` random seed to control the initialization.
- `maxSIM` the maximum number of CAVI iteration to perform.
- `warmstart` logical, if TRUE, the observational means of the cluster atoms are initialized with a k-means algorithm.
- `verbose` logical, if TRUE the iterations are printed.
Details

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 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 Gaussian likelihood, where both the mean and the variance are observational-cluster-specific, i.e.,

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

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

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

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

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

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

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

Value

`variational_CAM` returns a list of class `SANvb` containing 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 (optimized variational parameters). Details below.
- `time`: total computation time.

Data and parameters:

- `y`, `group`, `Nj`, `J`: Data, group labels, group frequencies, and number of groups.
- `K`, `L`: Number of fitted distributional and observational clusters.
m0, tau0, lambda0, gamma0  Model hyperparameters.
epsilon, seed  The threshold controlling the convergence criterion and the random seed adopted
to replicate the run.
(hyp_alpha1, hyp_alpha2) or alpha  Hyperparameters on $\alpha$ (if $\alpha$ random); or provided value for
$\alpha$ (if fixed).
(hyp_beta1, hyp_beta2) or beta  Hyperparameters on $\beta$ (if $\beta$ random); or provided value for $\beta$ (if
fixed).

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

theta_l  Matrix of size (L,4). Each row is a posterior variational estimate of the four normal-
inverse gamma hyperparameters.
Elbo_val  Vector containing the values of the ELBO.
XI  A list of length J. Each element is a matrix of size (N, L) posterior variational probability
of assignment of the i-th observation in the j-th group to the l-th OC, i.e.,
$\hat{\xi}_{i,j,l} = \bar{Q}(M_{i,j} = l)$.
RHO  Matrix of size (J, K). Each row is a posterior variational probability of assignment of the j-th
group to the k-th DC, i.e., $\hat{\rho}_{j,k} = \bar{Q}(S_j = k)$.

a_tilde_k, b_tilde_k  Vector of updated variational parameters of the Beta distributions govern-
ing the distributional stick-breaking process.

a_tilde_lk, b_tilde_lk  Matrix of updated variational parameters of the Beta distributions gov-
erning the observational stick-breaking process (arranged by column).

conc_hyper  If the concentration parameters are chosen to be random, these object contain a vector
with the four updated hyperparameters.
alpha,beta  If the concentration parameters are chosen to be fixed, these objects contain the passed
values.

References

Bayesian Nonparametric Analysis of Nested Data. Journal of the American Statistical Association,
118(541), 405-416. DOI: 10.1080/01621459.2021.1933499


Examples

```
set.seed(123)
y <- c(rnorm(100),rnorm(100,5))
g <- rep(1:2,rep(100,2))
est <- variational_CAM(y, g, verbose = FALSE, epsilon = 1e-2)
```
variational_fiSAN  
Mean Field Variational Bayes estimation of fiSAN

Description

variational_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 finite Dirichlet mixture at the observational one.

Usage

```r
variational_fiSAN(y, group,
  maxL = 30, maxK = 20,
  m0 = 0, tau0 = .01, lambda0 = 3, gamma0 = 2,
  conc_hyperpar = c(1,1), conc_par = NULL,
  beta_bar = .005,
  epsilon = 1e-6, seed = NULL,
  maxSIM = 1e5, warmstart = TRUE, verbose = FALSE)
```

Arguments

- **y**: Numerical vector of observations (required).
- **group**: Numerical vector of the same length of y, indicating the group membership (required).
- **maxL, maxK**: integers, the upper bounds for the observational and distributional clusters to fit, respectively.
- **m0, tau0, lambda0, gamma0**: Hyperparameters on \((\mu, \sigma^2) \sim NIG(m_0, \tau_0, \lambda_0, \gamma_0)\).
- **conc_hyperpar, conc_par**: Vectors of values used the concentration parameters of of the stick-breaking representation for the distributional and observational DPs, respectively. The following two arguments can be passed. Specifically,\n- **conc_hyperpar**: a vector with 2 positive entries: \((s_1, s_2)\). If a random concentration parameter \(\alpha\) is adopted, the specification is \(\alpha \sim Gamma(s_1, s_2)\). Default set to unitary vector.
- **conc_par**: a vector with one positive entry \(\alpha\). Default is set to NULL. If specified, the previous argument is ignored and the two concentration parameters are assumed fixed and equal to \(\alpha\).
- **beta_bar**: the hyperparameter of the symmetric observational Dirichlet distribution.
- **epsilon**: the tolerance that drives the convergence criterion adopted as stopping rule.
- **seed**: random seed to control the initialization.
- **maxSIM**: the maximum number of CAVI iteration to perform.
- **warmstart**: logical, if TRUE, the observational means of the cluster atoms are initialized with a k-means algorithm.
- **verbose**: logical, if TRUE the iterations are printed.
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 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

\[
Pr(S_j = k \mid \ldots) = \pi_k \quad \text{for } k = 1, 2, \ldots
\]

The distribution of the probabilities is \( \{\pi_k\}_{k=1}^{\infty} \sim GEM(\alpha) \), where GEM is the Griffiths-Engen-McCloskey distribution of parameter \( \alpha \), 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

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

The distribution of the probabilities is \( (\omega_{1,k}, \ldots, \omega_{L,k}) \sim Dirichlet_L(\beta/L, \ldots, \beta/L) \) for all \( k = 1, 2, \ldots \). Here, the dimension \( L \) is fixed.

Value

\texttt{variational_fiSAN} returns a list of class \texttt{SANvb} containing four objects:

- \texttt{model}: name of the fitted model.
- \texttt{params}: list containing the data and the parameters used in the simulation. Details below.
- \texttt{sim}: list containing the simulated values (optimized variational parameters). Details below.
- \texttt{time}: total computation time.

Data and parameters: \texttt{params} is a list with the following components:

- \( y \), \( \text{group} \), \( N_j \), \( J \) Data, group labels, group frequencies, and number of groups.
K, L  Number of fitted distributional and observational clusters.
m0, tau0, lambda0, gamma0  Model hyperparameters.
epsilon, seed  The threshold controlling the convergence criterion and the random seed adopted
to replicate the run.
(hyp_alpha1, hyp_alpha2) or alpha  Hyperparameters on \( \alpha \) (if \( \alpha \) random); or provided value for
\( \alpha \) (if fixed).
beta_bar  the hyperparameter governing all the finite Dirichlet distributions at the observational
level.

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

theta_l  Matrix of size (L,4). Each row is a posterior variational estimate of the four normal-
inverse gamma hyperparameters.
Elbo_val  Vector containing the values of the ELBO.
XI  A list of length J. Each element is a matrix of size (N, L) posterior variational probability
of assignment of the i-th observation in the j-th group to the l-th OC, i.e.,
\( \xi_{i,j,l} = Q(M_{i,j} = l) \).
RHO  Matrix of size (J, K). Each row is a posterior variational probability of assignment of the j-th
group to the k-th DC, i.e., \( \rho_{j,k} = Q(S_j = k) \).
a_tilde_k, b_tilde_k  Vector of updated variational parameters of the Beta distributions govern-
ing the distributional stick-breaking process.
beta_bar_lk  Matrix of updated variational parameters of the Dirichlet distributions governing the
observational clustering (arranged by column).
conc_hyper  If the concentration parameters is chosen to be random, these object contain a vector
with the two updated hyperparameters.
alpha  If the concentration parameters is chosen to be fixed, this object contains the passed values.

Examples

```r
set.seed(1234)
y <- c( rnorm(100), rnorm(100, 5))
g <- rep( 1:2, rep(100, 2))
est <- variational_fSAN( y, g, verbose = FALSE, epsilon = 1e-2)
```

variational_fSAN  Mean Field Variational Bayes estimation of fSAN

Description

variational_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 finite Dirichlet mixtures for both the distributional and observational levels of the model.
variational_fSAN

Usage
variational_fSAN(y, group, maxL = 30, maxK = 20,
m0 = 0, tau0 = .01, lambda0 = 3, gamma0 = 2,
alpha_bar = .005, beta_bar = .005,
epsilon = 1e-6, seed = NULL, maxSIM = 1e5,
warmstart = TRUE, verbose = FALSE)

Arguments
- **y**: Numerical vector of observations (required).
- **group**: Numerical vector of the same length of **y**, indicating the group membership (required).
- **maxL, maxK**: integers, the upper bounds for the observational and distributional clusters to fit, respectively.
- **m0, tau0, lambda0, gamma0**: Hyperparameters on \((\mu, \sigma^2) \sim NIG(m_0, \tau_0, \lambda_0, \gamma_0)\).
- **alpha_bar**: the hyperparameter of the symmetric distributional Dirichlet distribution.
- **beta_bar**: the hyperparameter of the symmetric observational Dirichlet distribution.
- **epsilon**: the tolerance that drives the convergence criterion adopted as stopping rule.
- **seed**: random seed to control the initialization.
- **maxSIM**: the maximum number of CAVI iteration to perform.
- **warmstart**: logical, if TRUE, the observational means of the cluster atoms are initialized with a k-means algorithm.
- **verbose**: logical, if TRUE the iterations are printed.

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 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

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

The distribution of the probabilities is $(\pi_1, \ldots, \pi_K) \sim \text{Dirichlet}_K(\alpha/K, \ldots, \alpha/K)$. Here, the dimension $K$ is fixed.

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

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

The distribution of the probabilities is $(\omega_{1,k}, \ldots, \omega_{L,k}) \sim \text{Dirichlet}_L(\beta/L, \ldots, \beta/L)$ for all $k = 1, \ldots, K$. Here, the dimension $L$ is fixed.

Value

`variational_fSAN` returns a list of class `SANvb` containing 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 (optimized variational parameters). Details below.
- `time`: total computation time.

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

- `y`, `group`, `Nj`, `J` Data, group labels, group frequencies, and number of groups.
- `K`, `L` Number of fitted distributional and observational clusters.
- `m0`, `tau0`, `lambda0`, `gamma0` Model hyperparameters.
- `epsilon`, `seed` The threshold controlling the convergence criterion and the random seed adopted to replicate the run.
- `alpha_bar`, `beta_bar` the hyperparameters governing all the finite Dirichlet distributions at the distributional and observational level.

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

- `theta_l` Matrix of size $(L, 4)$. Each row is a posterior variational estimate of the four normal-inverse gamma hyperparameters.
- `Elbo_val` Vector containing the values of the ELBO.
- `XI` A list of length $J$. Each element is a matrix of size $(N, L)$ posterior variational probability of assignment of the $i$-th observation in the $j$-th group to the $l$-th OC, i.e., $\hat{\xi}_{i,j,l} = \hat{Q}(M_{i,j} = l)$.
- `RHO` Matrix of size $(J, K)$. Each row is a posterior variational probability of assignment of the $j$-th group to the $k$-th DC, i.e., $\hat{\rho}_{j,k} = \hat{Q}(S_j = k)$.
- `a_tilde_k`, `b_tilde_k` Vector of updated variational parameters of the Beta distributions governing the distributional stick-breaking process.
- `alpha_bar_k` Vector of updated variational parameters of the Dirichlet distributions governing the distributional clustering.
- `beta_bar_lk` Matrix of updated variational parameters of the Dirichlet distributions governing the observational clustering (arranged by column).
References


Examples

```r
set.seed(123)
y <- c(rnorm(50),rnorm(50,5))
g <- rep(1:2,rep(50,2))
est <- variational_fSAN(y, g, verbose = FALSE,
                         epsilon = 1e-2, maxL=15, maxK=10)
```

---

**variational_multistart**

*Perform variational inference using multiple starting points.*

Description

`variational_multistart` is the main function of the package. It is used to estimate via variational inference the three models we present (CAM, fISAN, fSAN) while adopting multiple random starting points to better explore the variational parameter space. The run that provides the highest Expected Lower BOund (ELBO) is usually the one considered for inference. Note that the arguments passed to this function are a union of the arguments of the functions `variational_CAM`, `variational_fiSAN`, and `variational_fsAN`.

Usage

```r
variational_multistart(y, group, runs, cores = 1,
                        model = c("fiSAN","CAM","fSAN"),
                        maxL = 30, maxK = 20,
                        m0 = 0, tau@0 = .01, lambda@0 = 3, gamma@0 = 2,
                        conc_par = NULL, conc_hyperpar = c(1,1,1,1),
                        alpha_bar = 0.05, beta_bar = 0.05,
                        epsilon = 1e-6, root = 1234, maxSIM = 1e5,
                        warmstart = TRUE)
```

```r
## S3 method for class 'multistart'
plot(x, type = c("elbo", "time"), log_scale_iter = TRUE, ...)
```

```r
## S3 method for class 'multistart'
print(x, ...)```
Arguments

- **y**: vector of observations.
- **group**: vector of the same length of `y` indicating the group membership (numeric).
- **runs**: the number of multiple runs to launch.
- **cores**: the number of cores to dedicate to the multiple runs.
- **model**: a string specifying the model to use. It can be “fiSAN”, “CAM”, or “fSAN”.
- **maxL, maxK**: integers, the upper bounds for the observational and distributional clusters to fit, respectively.
- **m0, tau0, lambda0, gamma0**: hyperparameters on \((\mu, \sigma^2) \sim NIG(m_0, \tau_0, \lambda_0, \gamma_0)\).
- **conc_hyperpar, conc_par, alpha_bar, beta_bar**: these values crucially depend on the chosen model. See `variational_CAM`, `variational_fiSAN`, `variational_fSAN` for proper explanations.
- **epsilon**: the tolerance that drives the convergence criterion adopted as stopping rule.
- **root**: common part of the random seeds used to control the initialization in order to provide reproducibility even in paralleled settings.
- **maxSIM**: the maximum number of CAVI iteration to perform.
- **warmstart**: logical, if TRUE, the observational means of the cluster atoms are initialized with a k-means algorithm.
- **x**: an object of class `multistart`, obtained from the `variational_multistart` function.
- **type**: a string specifying the type of plot. It can be either “elbo” or “time”. The former displays the elbo trajectories, highlighting the best run. The latter provides a summary of the computational times.
- **log_scale_iter**: logical. If TRUE, when plotting the elbo trajectories, the x-axes is displayed in log-scale, enhancing the visualization of the results.
- **...**: ignored

Details

For the details of the single models, see their specific documentations: `variational_CAM`, `variational_fiSAN`, and `variational_fSAN`.

Value

A list with all the runs performed. Each element of the list is a fitted variational model of class `SANvb`.

See Also

`variational_CAM`, `variational_fiSAN`, `variational_fSAN`, `extract_best`
Examples

# Generate example data
set.seed(123)
y <- c(rnorm(100), rnorm(100, 5))
g <- rep(1:2, rep(100, 2))

# Estimate multiple models via variational inference
est <- variational_multistart(y, g, runs=5)
Index

estimate_atoms_weights_vi, 2, 2, 4
estimate_clustering_vi, 3
extract_best, 3, 4, 5, 16

plot.multistart
  (variational_multistart), 15
plot.SANvb, 5
plot.vi_atoms_weights
  (estimate_atoms_weights_vi), 2
plot.vi_clustering
  (estimate_clustering_vi), 3
print.multistart
  (variational_multistart), 15
print.SANvb, 6, 6
print.vi_atoms_weights
  (estimate_atoms_weights_vi), 2
print.vi_clustering
  (estimate_clustering_vi), 3

variational_CAM, 2–4, 6, 7, 16
variational_fiSAN, 2–4, 6, 10, 16
variational_fSAN, 2–4, 6, 12, 16
variational_multistart, 15