Package ‘AntMAN’

October 28, 2019

Version 1.0
Date 2019-10-07
Type Package
Title Anthology of Mixture Analysis Tools
Author Raffaele Argiento [aut],
Bruno Bodin [aut, cre],
Maria De Iorio [aut]
Maintainer Bruno Bodin <bruno.bodin@yale-nus.edu.sg>
Description Fits finite Bayesian mixture models with random number of component. The MCMC algorithm implemented is based on point processes as proposed by Argiento and De Iorio (2019) <arXiv:1904.09733> and offers a more computational efficient alternative to reversible jump. Different mixture kernels can be specified: univariate Gaussian, univariate Poisson, univariate binomial, multivariate Gaussian, multivariate Bernoulli (latent class analysis). For the parameters characterising the mixture kernel, we specify conjugate priors, with possibly user specified hyper-parameters. We allow for different choices for the prior on the number of components: shifted Poisson, negative binomial, and point masses (i.e. mixtures with fixed number of components).
License MIT + file LICENSE
LazyData true
URL https://github.com/bbodin/AntMAN
Imports stats, graphics, grDevices, Rcpp (>= 0.12.3), sdols, mvtnorm, mclust
Suggests dendextend, ggdendro, ggplot2, jpeg
LinkingTo Rcpp, RcppArmadillo
RoxygenNote 6.1.1
Encoding UTF-8
NeedsCompilation yes
Repository CRAN
Date/Publication 2019-10-28 15:20:06 UTC
R topics documented:

- AM_clustering_estimation_average
- AM_clustering_estimation_laugreen
- AM_clustering_estimation_squared_loss
- AM_coclustering
- AM_coclustering_slow
- AM_compute_stirling_ricor_abs
- AM_compute_stirling_ricor_log
- AM_emp_bayes_uninorm
- AM_find_gamma_Delta
- AM_find_gamma_NegBin
- AM_find_gamma_Pois
- AM_mcmc_fit
- AM_mcmc_output
- AM_mcmc_parameters
- AM_mix_components_prior_dirac
- AM_mix_components_prior_negbin
- AM_mix_components_prior_pois
- AM_mix_weights_prior_gamma
- AM_multiber_mix_hyperparams
- AM_multinorm_mix_hyperparams
- AM_plot_similarity_matrix
- AM_prior_K_Delta
- AM_prior_K_NegBin
- AM_prior_K_Pois
- AM_unibin_mix_hyperparams
- AM_uninorm_mix_hyperparams
- AM_unipois_mix_hyperparams
- AM_VnkDelta
- AM_VnkNegBin
- AntMAN
- brain
- carcinoma
- galaxy
- plot.AM_mcmc_output
- said
- summary.AM_mcmc_output
**AM_clustering_estimation_average**

*Return maximum likelihood estimation (average)*

Given a MCMC output, this function return maximum likelihood estimation.

**Description**

Return maximum likelihood estimation (average)
Given a MCMC output, this function return maximum likelihood estimation.

**Usage**

`AM_clustering_estimation_average(fit)`

**Arguments**

- `fit` [a `AM_mcmc_output` object](#)

**Value**

maximum likelihood estimation (average)

---

**AM_clustering_estimation_laugreen**

*Return maximum likelihood estimation (laugreen)*

Given a MCMC output, this function return maximum likelihood estimation.

**Description**

Return maximum likelihood estimation (laugreen)
Given a MCMC output, this function return maximum likelihood estimation.

**Usage**

`AM_clustering_estimation_laugreen(fit, C = NULL)`

**Arguments**

- `fit` [a `AM_mcmc_output` object](#)
- `C` used to speed up the function, can be the coclustering as returned from the command `AM_coclustering`.

**Value**

maximum likelihood estimation (laugreen)
AM_clustering_estimation_squared_loss

Return maximum likelihood estimation (squared_loss) Given a MCMC output, this function return maximum likelihood estimation.

Description

Return maximum likelihood estimation (squared_loss)
Given a MCMC output, this function return maximum likelihood estimation.

Usage

AM_clustering_estimation_squared_loss(fit)

Arguments

fit a AM_mcmc_output object

Value

maximum likelihood estimation (squared_loss)

AM_coclustering

Return co-clustering Given a MCMC output, this function return co-clustering matrix

Description

Return co-clustering
Given a MCMC output, this function return co-clustering matrix

Usage

AM_coclustering(fit)

Arguments

fit a AM_mcmc_output object

Value

coclustering matrix
AM_coclustering_slow

Return co-clustering slowly
Given a MCMC output, this function return co-clustering matrix

Description
Return co-clustering slowly
Given a MCMC output, this function return co-clustering matrix

Usage
AM_coclustering_slow(fit)

Arguments
fit a AM_mcmc_output object

Value
co-clustering matrix

AM_compute_stirling_ricor_abs

Compute the logarithm of the absolute value of the generalized Sriling number of second Kind (mi pare) See charambeloides, using a recursive formula Devo vedere la formula

Description
There are no default values.

Usage
AM_compute_stirling_ricor_abs(n, gamma)

Arguments
n The sample size
gamma A positive real number gamma

Value
A vector of length n, reporting the values C(gamma, n, k) for k=1,...,n

Examples
dd = AM_compute_stirling_ricor_abs(11, 10)
print(dd)
**Description**

There are no default values.

**Usage**

\[
AM\_compute\_stirling\_ricor\_log(n, \gamma)
\]

**Arguments**

- \(n\): The sample size
- \(\gamma\): A positive real number

**Value**

A vector of length \(n\), reporting the values \(\log(\gamma_k)\) for \(k=1,\ldots,n\)

**Examples**

```r
dd = AM_compute_stirling_ricor_log(11, 10)
print(dd)
```

**Description**

this function compute the hyperparameters of an Normal-Inverse-Gamma distribution using an empirical Bayes approach.

**Usage**

\[
AM\_emp\_bayes\_uninorm(y, scEmu = 1, scEsig2 = 3, CVsig2 = 3)
\]

**Arguments**

- \(y\): The data \(y\)
- \(scEmu\): a positive value \(scEmu\) (default =1) such that marginally \(E(\mu) = (\text{sample variance})\times scEmu\)
- \(scEsig2\): a positive value \(scEsig2\) (default=3) such that marginally \(E(\sigma^2) = (\text{sample variance})\times scEsig2\)
- \(CVsig2\): The coefficient of variation of \(\sigma^2\) (default =3), \(CVsig2\)
Value

Parameters of the normal inverse Gamma prior according to the empirical Bayes approach.

Once specified a fixed value of components $M^\ast$ this function adopt a bisection method to find the value of $\gamma$ such that the induced distribution on the number of clusters is centered around a user specified value $K^\ast$, i.e. the function use a bisection method to solve Eq.~eq:findgamma of WE NEED TO CITE ANTMAN PAPER. The user can provide a lower $\gamma_l$ and an upper $\gamma_u$ bound for the possible values of $\gamma$. The default values are $\gamma_l = 10^{-3}$ and $\gamma_u = 10$. A default value for the tolerance is $\epsilon = 0.1$. Moreover, after a maximum number of iteration (default is 31), the function stops warning that convergence has not bee reached.

Description

Once specified a fixed value of components $M^\ast$ this function adopt a bisection method to find the value of $\gamma$ such that the induced distribution on the number of clusters is centered around a user specified value $K^\ast$, i.e. the function use a bisection method to solve Eq.~eq:findgamma of WE NEED TO CITE ANTMAN PAPER. The user can provide a lower $\gamma_l$ and an upper $\gamma_u$ bound for the possible values of $\gamma$. The default values are $\gamma_l = 10^{-3}$ and $\gamma_u = 10$. A default value for the tolerance is $\epsilon = 0.1$. Moreover, after a maximum number of iteration (default is 31), the function stops warning that convergence has not bee reached.

Usage

\begin{verbatim}
AM_find_gamma_Delta(n, Mstar, Kstar = 6, gam_min = 1e-04,
                 gam_max = 10, tolerance = 0.1)
\end{verbatim}

Arguments

\begin{verbatim}
n               sample size
Mstar           number of component of the mixture
Kstar           mean number of cluster the user want to specify
gam_min         lower bound of the interval in which gamma should be lie (default 1e-4)
gam_max         upper bound of the interval in which gamma should lie (default 10)
tolerance       tolerance for the method
\end{verbatim}

Value

A value of $\gamma$ such that $E(K)=K^\ast$
Examples

```r
n <- 82
Mstar <- 12
gam_de <- AM_find_gamma_Delta(n,Mstar,Kstar=6, gam_min=1e-4,gam_max=10, tolerance=0.1)
prior_K_de <- AM_prior_K_Delta(n,gam_de,Mstar)
prior_K_de%*%1:n
```

**AM_find_gamma_NegBin**  
Once the prior on the number of mixture $M$ is assumed to be a Negative Binomial with parameter $r>0$ and $0<p<1$, with mean is $1 + r*p/(1-p)$, this function adopt a bisection method to find the value of $\gamma$ such that the induced distribution on the number of clusters is centered around a user specified value $K^*$, i.e. the function use a bisection method to solve Eq.~eq:findgamma of WE NEED TO CITE ANTMAN PAPER. The user can provide a lower $\gamma_l$ and an upper $\gamma_u$ bound for the possible values of $\gamma$. The default values are $\gamma_l = 10^{-3}$ and $\gamma_u = 10$. A default value for the tolerance is $\epsilon = 0.1$. Moreover, after a maximum number of iteration (default is 31), the function stops warning that convergence has not bee reached.

**Description**

Once the prior on the number of mixture $M$ is assumed to be a Negative Binomial with parameter $r>0$ and $0<p<1$, with mean is $1 + r*p/(1-p)$, this function adopt a bisection method to find the value of $\gamma$ such that the induced distribution on the number of clusters is centered around a user specified value $K^*$, i.e. the function use a bisection method to solve Eq.~eq:findgamma of WE NEED TO CITE ANTMAN PAPER. The user can provide a lower $\gamma_l$ and an upper $\gamma_u$ bound for the possible values of $\gamma$. The default values are $\gamma_l = 10^{-3}$ and $\gamma_u = 10$. A default value for the tolerance is $\epsilon = 0.1$. Moreover, after a maximum number of iteration (default is 31), the function stops warning that convergence has not bee reached.

**Usage**

```
AM_find_gamma_NegBin(n, r, p, Kstar = 6, gam_min = 0.001, gam_max = 10000, tolerance = 0.1)
```

**Arguments**

- **n**: The sample size
- **r**: The dispersion parameter $r$ of Negative Binomial
- **p**: The probability of failure parameter $p$ of Negative Binomial
- **Kstar**: The mean number of cluster the user want to specify
- **gam_min**: The lower bound of the interval in which gamma should be lie
- **gam_max**: The upper bound of the interval in which gamma should lie
- **tolerance**: tolerance of the method
**Value**

A value of gamma such that \(E(K) = K^*\)

**Examples**

```r
n <- 82
r <- 1
p <- 0.8571

gam_nb = AM_find_gamma_NegBin(n, r, p, Kstar = 6, gam_min = 1e-04, gam_max = 10, tolerance = 0.1)
prior_K_nb = AM_prior_K_NegBin(n, gam_nb, r, p)
prior_K_nb %*% 1:n
```

**Description**

Once the prior on the number of mixture \(M\) is assumed to be a Shifted Poisson of parameter Lambda, this function adopt a bisection method to find the value of gamma such that the induced distribution on the number of clusters is centered around a user specified value \(K^*\), i.e. the function use a bisection method to solve Eq.~eq:findgamma of WE NEED TO CITE ANTMAN PAPER. The user can provide a lower \(\gamma_l\) and an upper \(\gamma_u\) bound for the possible values of \(\gamma\). The default values are \(\gamma_l = 10^{-3}\) and \(\gamma_u = 10\). A default value for the tolerance is \(\epsilon = 0.1\). Moreover, after a maximum number of iteration (default is 31), the function stops warning that convergence has not been reached.

**Usage**

```r
AM_find_gamma_Pois(n, Lambda, Kstar = 6, gam_min = 1e-04, gam_max = 10, tolerance = 0.1)
```

**Arguments**

- `n`: The sample size
- `Lambda`: The parameter of the Shifted Poisson for the number of components of the mixture
- `Kstar`: The mean number of cluster the user want to specify
The lower bound of the interval in which \( \gamma \) should lie

\( \gamma_{\text{max}} \)

The upper bound of the interval in which \( \gamma \) should lie

\( \text{tolerance} \)

Tolerance of the method

**Value**

A value of \( \gamma \) such that \( E(K) = K^* \)

**Examples**

```r
n <- 82
Lam <- 11
gam_po <- AM_find_gamma_Pois(n,Lam,Kstar=6, gam_min=0.0001,gam_max=10, tolerance=0.1)
prior_K_po <- AM_prior_K_Pois(n,gam_po,Lam)
prior_K_po%*%1:n
```

**Description**

The `AM_mcmc_fit` function performs a Gibbs sampling in order to estimate a mixture of a predefined type `mix_kernel_hyperparams` (defined with `AM_*_mix_hyperparams` functions, where `*` denotes the chosen kernel) sample data \( y \). Additionally, a prior distribution on the number of mixture components must be specified through `mix_components_prior` (generated with `AM_mix_components_prior_*` functions, where `*` denotes the chosen prior). Similarly, a prior on the weights of the mixture is specified through `mix_weight_prior` (defined with `AM_mix_weights_prior_*` functions). Finally, with `mcmc_parameters` the user sets the MCMC parameters for the Gibbs sampler (defined with `AM_mcmc_parameters` functions).

**Usage**

```r
AM_mcmc_fit(y, mix_kernel_hyperparams, initial_clustering = NULL, 
             init_K = NULL, fixed_clustering = NULL, 
             mix_components_prior = AM_mix_components_prior_pois(), 
             mix_weight_prior = AM_mix_weights_prior_gamma(), 
             mcmc_parameters = AM_mcmc_parameters())
```

**Arguments**

- `y`: input data, can be a vector or a matrix.
- `mix_kernel_hyperparams`: is a configuration list, defined by `*_mix_hyperparams` functions, where `*` denotes the chosen kernel.
- `initial_clustering`: is a vector CI of initial cluster assignment. If no clustering is specified (either as `init_K` or `init_clustering`), then every observation is assigned to its own cluster.
init_K  initial value for the number of cluster. When this is specified, AntMAN initialises the clustering assign using K-means.

fixed_clustering  is a vector CI of cluster assignment that will remained unchanged for every iterations.

mix_components_prior  is a configuration list defined by AM_mix_components_prior_* functions, where * denotes the chosen prior.

mix_weight_prior  is a configuration list defined by AM_weight_prior_* functions, where * denotes the chosen prior specification.

mcmc_parameters  is a configuration list defined by AM_mcmc_parameters.

Details
If no initial clustering is specified (either as init_K or init_clustering), then every observation is allocated to a different cluster. If init_K is specified then AntMAN initialises the clustering through K-means.

Warning: if the user does not specify init_K or initial_cluster, the first steps can be time-consuming because of default value for the initial clustering.

Value
The return value is an AM_mcmc_output object.

Examples
AM_mcmc_fit( AM_sample_unipois()$y,
AM_unipois_mix_hyperparams (alpha0=2, beta0=0.2),
mcmc_parameters = AM_mcmc_parameters(niter=200, burnin=100, thin=10))

AM_mcmc_output  S3 class AM_mcmc_output.

Description
Output type of return values from AM_mcmc_fit. See paper for the moment.

See Also
AM_mcmc_fit
AM_mcmc_parameters

MCMC Parameters

Description

This function generates an MCMC parameters list to be used as mcmc_parameters argument within AM_mcmc_fit.

Usage

AM_mcmc_parameters(niter = 5000, burnin = 2500, thin = 1, verbose = 1, output = c("CI", "K"), parallel = TRUE, output_dir = NULL)

Arguments

- niter: Total number of MCMC iterations.
- burnin: Number of iterations to discard as burn-in.
- thin: Thining rate.
- verbose: A value from 0 to 4, that specifies the desired level of verbosity (0:None, 1:Warnings, 2:Debug, 3:Extras)
- output: A list of parameters output to return
- parallel: Some of the algorithms can be run in parallel using OpenMP. When set to True, this parameter triggers the parallelism.
- output_dir: Path to an output dir, where to store all the outputs.

Details

The niter argument specify the total number of iteration. burnin is the number of initial iterations to discard. thin specifies how often a draw from the posterior distribution is stored after burnin, i.e. one every -th samples is saved. Therefore, the total number of MCMC samples saved is (niter -burnin)/thin. If thin =1, then AntMAN stores every iteration.

Value

list to be used as mcmc_parameters argument with AM_mcmc_fit.

Examples

AM_mcmc_parameters (niter=1000, burnin=10000, thin=50)
AM_mcmc_parameters (niter=1000, burnin=10000, thin=50, output=c("CI","S","TAU"))
**AM_mix_components_prior_dirac**

Generate a configuration object that contains a Point mass prior.

**Description**

Generate a configuration object that assigns a Point mass prior to the number of mixture components. This is the simplest option and it requires to specify a value $\tilde{M}$ such that $Pr(M = \tilde{M}) = 1$.

**Usage**

```
AM_mix_components_prior_dirac(Mstar)
```

**Arguments**

- **Mstar**: Fixed value $\tilde{M}$ for the number of components.

**Value**

list to be used as `mix_components_prior` argument for `AM_mcmc_fit`.

**See Also**

- `AM_mcmc_fit`

**Examples**

```
AM_mix_components_prior_dirac (Mstar=3)
```

**AM_mix_components_prior_negbin**

Negative Binomial Prior.

**Description**

This generate a configuration object for a Shifted Negative Binomial prior on the number of mixture components such as

$$q_M(m) = Pr(M = m) = \frac{\Gamma(r + m - 1)}{(m - 1)!\Gamma(r)} p^{m-1} (1 - p)^r, \quad m = 1, 2, 3, \ldots$$

The hyper-parameters $p \in (0, 1)$ (probability of success) and $r > 0$ (size) can either be fixed using $r$ and $p$ or assigned appropriate prior distributions. In the latter case, we assume $p \sim Beta(a_P, b_P)$ and $r \sim Gamma(a_R, b_R)$. In AntMAN we assume the following parametrization of the Gamma density:

$$p(x | a, b) = \frac{b^a x^{a-1}}{\Gamma(a)} \exp{-bx}, \quad x > 0$$
Usage

`AM_mix_components_prior_negbin(a_R = NULL, b_R = NULL, a_P = NULL, b_P = NULL, R = NULL, P = NULL, init_R = NULL, init_P = NULL)`

Arguments

- `a_R` The shape parameter `a` of the `Gamma(a,b)` prior distribution for `r`.
- `b_R` The rate parameter `b` of the `Gamma(a,b)` prior distribution for `r`.
- `a_P` The parameter `a` of the `Beta(a,b)` prior distribution for `p`.
- `b_P` The parameter `b` of the `Beta(a,b)` prior distribution for `p`.
- `R` It allows to fix `r` to a specific value.
- `P` It allows to fix `p` to a specific value.
- `init_R` The initial value of `r`, when specifying `a_R` and `b_R`.
- `init_P` The initial value of `p`, when specifying `a_P` and `b_P`.

Details

If no arguments are provided, the default is `r = 1, a_P = 1, b_P = 1`.

Additionally, when `init_R` and `init_P` are no specified, there is default values: `init_R = 1` and `init_P = 0.5`.

Value

A configuration list to be used as `mix_components_prior` argument for `AM_mcmc_fit`.

See Also

`AM_mcmc_fit`

Examples

`AM_mix_components_prior_negbin (R=1, P=1)`
`AM_mix_components_prior_negbin ()`
**AM_mix_components_prior_pois**

**Description**

This function generates a configuration object for a Shifted Poisson prior on the number of mixture components such that

\[
q_M(m) = Pr(M = m) = e^{-\Lambda}\Lambda^{m-1} / (m-1)!, \quad m = 1, 2, 3, \ldots
\]

The hyper-parameter \(\Lambda\) can either be fixed using \(\text{Lambda}\) or assigned a \(\text{Gamma}(a, b)\) prior distribution with \(a\) and \(b\). In AntMAN we assume the following parametrization of the Gamma density:

\[
p(x \mid a, b) = \frac{b^a x^{a-1}}{\Gamma(a)} \exp\{-bx\}, \quad x > 0
\]

**Usage**

```
AM_mix_components_prior_pois(a = NULL, b = NULL, Lambda = NULL, init = NULL)
```

**Arguments**

- **a**
  - The shape parameter \(a\) of the \(\text{Gamma}(a, b)\) prior distribution.
- **b**
  - The rate parameter \(b\) of the \(\text{Gamma}(a, b)\) prior distribution.
- **Lambda**
  - It allows to set the hyper-parameter \(\Lambda\) to fixed value.
- **init**
  - The initial value for \(\Lambda\), when specifying \(a\) and \(b\).

**Details**

If no arguments are provided, the default is a prior distribution with \(a = 1\) and \(b = 1\).

**Value**

A configuration list to be used as `mix_components_prior` argument for `AM_mcmc_fit`.

**See Also**

`AM_mcmc_fit`

**Examples**

```r
## See \code{\link{AM_uninorm_mix_hyperparams}} example.
components_prior = AM_mix_components_prior_pois (init=3, a=1, b=1)
```
AM_mix_weights_prior_gamma

Generate a configuration object to specify a prior on the hyper-parameter $\gamma$ for the Dirichlet prior on the mixture weights.

Description

Generate a configuration object to specify a prior on the hyper-parameter $\gamma$ for the Dirichlet prior on the mixture weights. We assume $\gamma \sim \text{Gamma}(a,b)$. Alternatively we can fix $\gamma$ to a specific value. Default is $\gamma = 1/N$, where N is the number of observations. In AntMAN we assume the following parametrization of the Gamma density:

$$p(x | a, b) = \frac{b^a x^{a-1}}{\Gamma(a)} \exp\{-bx\}, \quad x > 0$$

Usage

```r
AM_mix_weights_prior_gamma(a = NULL, b = NULL, gamma = NULL, init = NULL)
```

Arguments

- `a` The shape parameter $a$ of the Gamma prior
- `b` The rate parameter $b$ of the Gamma prior
- `gamma` It allows to fix $\gamma$ to a specific value.
- `init` The init value for $\gamma$, when we assume $\gamma$ random.

Value

A configuration list to be used as `mix_components_prior` argument for `AM_mcmc_fit`.

Examples

```r
AM_mix_weights_prior_gamma (a=1, b=1)  
AM_mix_weights_prior Gamma (a=1, b=1, init=1)  
AM_mix_weights_prior Gamma (gamma = 3)  
AM_mix_weights_prior Gamma ()
```
AM_multiber_mix_hyperparams

Multivariate Bernoulli Mixture Hyperparameters (Latent Class analysis)

Description

Generate a configuration object that defines the prior hyperparameters for a mixture of multivariate Bernoulli. If the dimension of the data is \( P \), then the prior is a product of \( P \) independent Beta distributions, \( \text{Beta}(a_0, a_0) \). Therefore, the vectors of hyperparameters, \( a_0 \) and \( b_0 \), are \( P \)-dimensional. Default is \( (a_0 = c(1, \ldots, 1), b_0 = c(1, \ldots, 1)) \)

Usage

\[
\text{AM_multiber_mix_hyperparams}(a0, b0)
\]

Arguments

- \( a0 \) The \( a0 \) hyperparameters.
- \( b0 \) The \( b0 \) hyperparameters.

Value

A list to be used as \( \text{mix_kernel_hyperparams} \) argument for \( \text{mcmc_fit} \).

Examples

\[
\text{AM_multiber_mix_hyperparams}(a0= c(1,1,1,1), b0= c(1,1,1,1))
\]

AM_multinorm_mix_hyperparams

Multivariate Normal Mixture Hyperparameters.

Description

This function allows the user to specify the hyperparameters for the conjugate prior for a mixture of Multivariate Normals. We assume that the data are \( d \)-dimensional vectors \( y_i \), where \( y_i \) are iid Normal random variables with mean \( \mu \) and covariance matrix \( \Sigma \). The conjugate prior is

\[
\pi(\mu, \Sigma \mid m_0, \kappa_0, \nu_0, \Lambda_0) = \pi_\mu(\mu \mid \Sigma, m_0, \kappa_0)\pi_\Sigma(\Sigma \mid \nu_0, \Lambda_0)
\]

\[
\pi_\mu(\mu \mid \Sigma, m_0, \kappa_0) = \frac{\sqrt{\kappa_0^d}}{\sqrt{(2\pi)^d} |\Sigma|} \exp \left( -\frac{\kappa_0}{2} (\mu - m_0)^T \Sigma^{-1} (\mu - m_0) \right), \quad \mu \in \mathbb{R}^d
\]

\[
\pi_\Sigma(\Sigma \mid \nu_0, \Lambda_0) = \frac{|\Lambda_0|^{\nu_0/2}}{2^{\nu_0 d/2} \Gamma(d/2)} |\Sigma|^{-(\nu_0 + d + 1)/2} e^{-\frac{1}{2} \text{tr}(\Lambda_0 \Sigma^{-1})}, \quad \Sigma^2 > 0
\]

with \( \mu \theta \) corresponds to \( m_0 \), \( \kappa \theta \) corresponds to \( \kappa_0 \), \( \nu \theta \) to \( \nu_0 \), \( \Lambda \theta \) to \( \Lambda_0 \).
Usage

\texttt{AM\_multinorm\_mix\_hyperparams(mu0 = NULL, ka0 = NULL, nu0 = NULL, Lam0 = NULL)}

Arguments

- \texttt{mu0}:
  - The hyperparameter \( m_0 \).
- \texttt{ka0}:
  - The hyperparameter \( \kappa_0 \).
- \texttt{nu0}:
  - The hyperparameter \( \nu_0 \).
- \texttt{Lam0}:
  - The hyperparameter \( \Lambda_0 \).

Details

Default is \((mu0=c(0,..,0), ka0=1, nu0=Dim+2, Lam0=diag(Dim))\) with \( Dim \) is the dimension of the data \( y \). We advise the user to set \( \nu_0 \) equal to at least the dimension of the data, \( Dim \), plus 2.

Value

A list to be used as \texttt{mix\_kernel\_hyperparams} argument for \texttt{mcmc\_fit}.

Examples

\texttt{AM\_multinorm\_mix\_hyperparams(())}

---

\texttt{AM\_plot\_similarity\_matrix}

\textit{Plot the Similarity Matrix Given a MCMC output, this function will produce an image of the Similarity Matrix}

Description

Plot the Similarity Matrix

Given a MCMC output, this function will produce an image of the Similarity Matrix

Usage

\texttt{AM\_plot\_similarity\_matrix(C, \ldots)}

Arguments

- \texttt{C}:
  - coclustering as returned from the command \texttt{AM\_coclustering}.
- \texttt{\ldots}:
  - All additional parameters will be passed to the image command.
This function computes the prior on the number of clusters, i.e., occupied components of the mixture for a Finite Dirichlet process when the prior on the component weights of the mixture is a Dirichlet with parameter \( \gamma \) (i.e., when unnormalized weights are distributed as \( \text{Gamma}(\gamma, 1) \) ) when the number of components are fixed to \( M^* \), i.e., a Dirac prior assigning mass only to \( M^* \) is assumed. See Section 9.1.1 of the Paper Argiento de Iorio 2019 for more details. There are no default values.

**Usage**

\[
\text{AM\_prior\_K\_Delta}(n, \gamma, M^*)
\]

**Arguments**

- **n**: The sample size
- **\( \gamma \)**: The gamma parameter of the Dirichlet
- **M^***: The number of components of the mixture

**Value**

A vector of length \( n \), reporting the values \( V(n, k) \) for \( k=1, \ldots, n \)

**Examples**

```r
n <- 82
\gamma_n <- 0.1743555
Mstar <- 12
prior_K_de <- AM\_prior\_K\_Delta(n, \gamma_n, Mstar)
plot(1:n, prior_K_de, type = "n", bty = "l", xlab = "k", ylab = "P(K=k)")
```
AM_prior_K_NegBin

This function compute the prior on the number of cluster, i.e. occupied component of the mixture for a Finite Dirichlet process when the prior on the component-weights of the mixture is a Dirichlet with parameter gamma (i.e. when unnormalized weights are distributed as Gamma(γ,1) ) when the prior on the number of component is Negative Binomial with parameter r>0 and 0<p<1, with mean is mu =1+ r*p/(1-p) TODO: CHECK THIS FORMULA!!!. See Section 9.1.1 of the Paper Argiento de Iorio 2019 for more details.

Description

There are no default values.

Usage

AM_prior_K_NegBin(n, gamma, r, p)

Arguments

n The sample size

gamma The gamma parameter of the Dirichlet

r The dispersion parameter r of Negative Binomial

p The probability of failure parameter p of Negative Binomial

Value

A vector of length n, reporting the values V(n,k) for k=1,...,n

Examples

n <- 50
gamma <- 1
r <- 0.1
p <- 0.91
gam_nb <- 0.2381641
prior_K_nb <- AM_prior_K_NegBin(n,gam_nb,r,p)
plot(1:n,prior_K_nb, type = "n", bty = "l", xlab = "k", ylab = "P(K=k)"
lines(1:n,prior_K_nb,type="h",lwd=2)
This function computes the prior on the number of clusters, i.e., occupied component of the mixture for a finite Dirichlet process when the prior on the component weights of the mixture is a Dirichlet with parameter \( \gamma \) (i.e., when unnormalized weights are distributed as \( \text{Gamma}(\gamma, 1) \)) when the prior on the number of components is Shifted Poisson of parameter \( \Lambda \). See Section 9.1.1 of Argiento de Iorio (2019) for more details.

**Description**

There are no default values.

**Usage**

```r
AM_prior_K_Pois(n, gamma, Lambda)
```

**Arguments**

- `n`: The sample size
- `gamma`: The \( \gamma \) parameter of the Dirichlet
- `Lambda`: The \( \Lambda \) parameter of the Poisson

**Value**

A vector of length \( n \), reporting the values of the prior on the number of clusters induced by the prior on \( M \) and \( w \), i.e., \( P(K) \) for \( k = 1, \ldots, n \). See Section 9.1.1 of Argiento de Iorio (2019) for more details.

**Examples**

```r
n <- 82
Lambda <- 10
gam_po <- 0.1550195
prior_K_po <- AM_prior_K_Pois(n, gam_po, Lambda)
plot(1:n, prior_K_po, type = "n", bty = "l", xlab = "k", ylab = "P(K=k)")
```
AM_unibin_mix_hyperparams

*Univariate Binomial Mixture Hyperparameters. Generate a configuration object that specifies the prior hyperparameters for a mixture of Univariate Binomial kernels with probability of success $\tau$ and size $N$. The conjugate prior on $\tau$ is a Beta distribution:*

$$
\pi(\tau | \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \tau^{\alpha - 1} (1 - \tau)^{\beta - 1}, \quad 0 \leq \tau \leq 1
$$

$N$ is fixed by the user and should always be specified. Here, $\alpha$ corresponds to $a_0$, $\beta$ to $b_0$. The default for the prior hyperparameters is $a_0 = 1, b_0 = 1$.

### Usage

```r
AM_unibin_mix_hyperparams(a0, b0, N)
```

### Arguments

- **a0**
  - The $a_0$ hyperparameter.
- **b0**
  - The $b_0$ hyperparameter.
- **N**
  - Size of the Binomial distribution.

### Value

A list to be used as `mix_kernel_hyperparams` argument for `mcmc_fit`.

### Examples

```r
AM_unibin_mix_hyperparams(a0=1, b0=1, N=100)
```
Univariate Normal Mixture Hyperparameters

Description

Generate a configuration object that specifies univariate Normal mixture kernel and allows to set the hyperparameters of the Normal-InverseGamma conjugate prior. As such, the kernel is a Gaussian distribution with mean $\mu$ and variance $\sigma^2$. The prior on $(\mu, \sigma^2)$ the Normal-InverseGamma:

$$
\pi(\mu, \sigma^2 | m_0, \kappa_0, \nu_0, \sigma_0^2) = \pi_\mu(\mu | \sigma^2, m_0, \kappa_0) \pi_{\sigma^2}(\sigma^2 | \nu_0, \sigma_0^2)
$$

$$
\pi_\mu(\mu | \sigma^2, m_0, \kappa_0) = \frac{\sqrt{\kappa_0}}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\kappa_0}{\sigma^2} (\mu - m_0)^2\right), \quad \mu \in \mathbb{R}
$$

$$
\pi_{\sigma^2}(\sigma^2 | \nu_0, \sigma_0^2) = \frac{\sigma_0^{2\nu_0}}{\Gamma(\nu_0)} (1/\sigma^2)^{\nu_0+1} \exp\left(-\frac{\sigma_0^2}{\sigma^2}\right), \quad \sigma^2 > 0
$$

Usage

`AM_uninorm_mix_hyperparams(m0, k0, nu0, sig02)`

Arguments

- `m0`: The $m_0$ hyperparameter.
- `k0`: The $\kappa_0$ hyperparameter.
- `nu0`: The $\nu_0$ hyperparameter.
- `sig02`: The $\sigma_0^2$ hyperparameter.

Details

where $m_0$ corresponds $m0$, $\kappa_0$ corresponds $k0$, $\nu_0$ corresponds $nu0$, $\sigma_0^2$ corresponds $sig02$.

If hyperparameters are not specified, the default is $m0=0, k0=1, nu0=3, sig02=1$.

Value

A list to be used as `mix_kernel_hyperparams` argument for `mcmc_fit`.

Examples

```r
#### This example ...
data(galaxy)
y_uvn = galaxy
mixture_uvn_params = AM_uninorm_mix_hyperparams (m0=20.83146, k0=0.3333333,
nu0=4.222222, sig02=3.661027)
```
mcmc_params = AM_mcmc_parameters(niter=200, burnin=50, thin=10, verbose=1)
components_prior = AM_mix_components_prior_pois (init=3, a=1, b=1)
weights_prior = AM_mix_weights_prior_gamma(init=2, a=1, b=1)

fit <- AM_mcmc_fit(
y = y_uvn,
mix_kernel_hyperparams = mixture_uvn_params,
mix_components_prior = components_prior,
mix_weight_prior = weights_prior,
mcmc_parameters = mcmc_params)

summary (fit)
plot (fit)

---

AM_unipois_mix_hyperparams

Univariate Poisson Mixture Hyperparameters.

Description

Generate a configuration object that specifies a univariate Poisson mixture kernel and allows to specify the hyperparameters of the conjugate Gamma prior, i.e. the kernel is a $\text{Poisson}(\tau)$ and $\tau \sim \text{Gamma}(\alpha_0, \beta_0)$. In AntMAN we assume the following parametrization of the Gamma density:

$$p(x \mid a, b) = \frac{b^a x^{a-1}}{\Gamma(a)} \exp\{-bx\}, \ x > 0$$

Usage

AM_unipois_mix_hyperparams(alpha0, beta0)

Arguments

- alpha0: The shape hyperparameter $\alpha_0$.
- beta0: The rate hyperparameter $\beta_0$.

Details

Note, by default alpha0=1 and beta0=1.

Value

A list to be used as \code{mix_kernel_hyperparams} argument for \code{mcmc_fit}.

Examples

AM_unipois_mix_hyperparams (alpha0=2, beta0=0.2)
Compute the value $V(n,k)$, needed to calculate the eppf of a Finite Dirichlet process when the prior on the component-weights of the mixture is a Dirichlet with parameter $\gamma$ (i.e. when unnormalised weights are distributed as Gamma$(\gamma,1)$) when the number of components are fixed to $M^*$, i.e. a Dirac prior assigning mass only to $M^*$ is assumed. See Section 9.1.1 of the Paper Argiento de Iorio 2019 for more details.

Description

There are no default values.

Usage

$AM\_VnkDelta(n, Mstar, gamma)$

Arguments

- $n$: The sample size
- $Mstar$: The number of component of the mixture
- $gamma$: The gamma parameter of the Dirichlet

Value

A vector of length $n$, reporting the values $V(n,k)$ for $k=1,\ldots,n$

Examples

```r
n=200
Mstar=100
gam=0.5
vvv=AM_VnkDelta(n,Mstar,gam);
stir= AM_compute_stirling_ricor_log(n, gam)
stir
plot(exp(vvv+stir) )
sum(exp(vvv+stir ) )
```
AM_VnkNegBin

Compute the value \(V(n,k)\), needed to calculate the epff of a Finite Dirichlet process when the prior on the component-weights of the mixture is a Dirichlet with parameter \(\gamma\) (i.e. when unnormalized weights are distributed as \(\text{Gamma}(\gamma,1)\)) when the prior on the number of component is Negative Binomial with parameter \(r\) and \(p\) with mean is \(\mu = 1 + r p / (1-p)\) TODO: CHECK THIS FORMULA!!!. See Section 9.1.1 of the Paper Argiento de Iorio 2019 for more details

Description

There are no default values.

Usage

\[
\text{AM_VnkNegBin}(n, r, p, \text{gam})
\]

Arguments

- \(n\): The sample size
- \(r\): The dispersion parameter \(r\) of Negative Binomial
- \(p\): The probability of failure parameter \(p\) of Negative Binomial
- \(\text{gam}\): The \(\gamma\) parameter of the Dirichlet

Value

A vector of length \(n\), reporting the values \(V(n,k)\) for \(k=1,\ldots,n\)

Examples

\[
\begin{align*}
n &= 1000 \\
r &= 1000 \\
p &= 0.5 \\
gam &= 0.5 \\
\text{vnk} &= \text{AM_VnkNegBin}(n, r, p, \text{gam}) \\
\text{stir} &= \text{AM_compute_stirling_ricor_log}(n, \text{gam}) \\
\text{plot}(\exp(\text{vnk}+\text{stir}+(1:n)\times\log(\text{gam}))) \\
\text{sum}(\exp(\text{vnk}+\text{stir}))
\end{align*}
\]
AM_VnkPoisson

Compute the value \( V(n,k) \), needed to calculate the eppf of a Finite Dirichlet process when the prior on the component-weights of the mixture is a Dirichlet with parameter \( \gamma \) (i.e. when unnormalized weights are distributed as \( \text{Gamma}(\gamma,1) \)) when the prior on the number of components is Shifted Poisson of parameter \( \Lambda \). See Section 9.1.1 of the Paper Argiento de Iorio 2019.

Description

There are no default values.

Usage

\[
\text{AM_VnkPoisson}(n, \Lambda, \gamma)
\]

Arguments

- \( n \) The sample size
- \( \Lambda \) The \( \Lambda \) parameter of the Poisson
- \( \gamma \) The \( \gamma \) parameter of the Dirichlet

Value

A vector of length \( n \), reporting the values \( V(n,k) \) for \( k=1, \ldots, n \)

Examples

\[
\begin{align*}
n &= 1000 \\
\Lambda &= 100 \\
\gamma &= 0.5 \\
vnk &= \text{AM_VnkPoisson}(n, \Lambda, \gamma) \\
stir &= \text{AM_compute_stirling_ricor_log}(n, \gamma) \\
\text{plot}(\exp(vnk+stir)) \\
\text{sum}(\exp(vnk+stir))
\end{align*}
\]

AntMAN

AntMAN: A package for fitting Finite Bayesian Mixture model with random number of component.
Description

AntMAN: Anthology of Mixture ANalysis tools AntMan is a R package to fit Finite Bayesian Mixture model with random number of component. The MCMC algorithm beyond AntMan is based on point processes and offer a more computational efficient alternative to Reversible Jump. Different mixture kernels can be specified: Univariate Gaussian, Univariate Poisson, Univariate Binomial, Multivariate Gaussian, Multivariate Bernoulli (Latent Class Analysis). For the parameters characterising the mixture kernel, we specify conjugate priors, with possibly user specified hyperparameters. We allow for different choices for the prior on the number of components: Shifted Poisson, Negative Binomial, and Point Masses (i.e. mixtures with fixed number of components).

Prior functions

The Prior functions ...

Package Philosophy

The main function of the AntMAN package is AM_mcmc_fit. AntMAN performs a Gibbs sampling in order to fit, in a Bayesian framework, a mixture model of a predefined type mix_kernel_hyperparams given a sample y. Additionally AntMAN allows the user to specify a prior on the number of components mix_components_prior and on the weights mix_weight_prior of the mixture. MCMC parameters mcmc_parameters need to be given as argument for the Gibbs sampler (number of interation, burn-in, ...). Initial values for the number of cluster (init_K) or a specific clustering allocation (init_clustering) can also be user-specify. Otherwise, by the default allocation we assign a different cluster for each element of the sample y as initial allocation. This choice can be computationally inefficient.

For example, in order to identify clusters over a population of patients given a set of medical assumptions:

mcmc = AM_mcmc_parameters(niter=20000)
mix = AM_multiber_mix_hyperparams ()
fit = AM_mcmc_fit (mix, mcmc)
summary (fit)

In this example AM_multiber_mix_hyperparams is one of the possible mixture to identify.

AntMAN currently support five different mixtures :

AM_unipois_mix_hyperparams(alpha0, beta0)
AM_uninorm_mix_hyperparams(m0, k0, nu0, sig02)
AM_unibin_mix_hyperparams(a0, b0, mb)
AM_multiber_mix_hyperparams(a0, b0)
AM_multinorm_mix_hyperparams(mu0, ka0, nu0, Lam0)

Additionnaly, there is three prior_component available :

AM_mix_components_prior_pois()
AM_mix_components_prior_negbin()
AM_mix_components_prior_dirac()
For example, in the context of image segmentation, where a maximal number of colour is require, a prior dirac can be used:

```python
mcmc = AM_mcmc_parameters(niter=20000)
mix = AM_multinorn_mix_hyperparams()
prior_component = AM_mix_components_prior_dirac(10)  # nothing more than 10 colours
fit = AM_mcmc_fit(mix, prior_component, mcmc) summary(fit)
```

---

### brain

**Teen Brain Images from the National Institutes of Health, U.S.**

**Description**

Picture of brain activities from a teenager consuming drugs.

**Usage**

```r
brain
```

**Format**

A list that contains `dim` (W:width,H:height) pair, and `pic` a data frame (W*H pixels image in RGB format).

**Source**

https://www.flickr.com/photos/nida-nih/29741916012

**References**


**Examples**

```r
data(brain)
```
carcinoma

The carcinoma data from Agresti (2002, 542) consist of seven dichotomous variables that represent the ratings by seven pathologists of 118 slides on the presence or absence of carcinoma. The purpose of studying these data is to model “interobserver agreement” by examining how subjects might be divided into groups depending upon the consistency of their diagnoses.

Usage

carcinoma

Format

A data frame with 118 rows and 7 variables (from A to G).

References


Examples

data(carcinoma)

galaxy

Galaxy velocities dataset

Description

This data set consider physical information on velocities (km/second) for 82 galaxies reported by Roeder (1990). These are drawn from six well-separated conic sections of the Corona Borealis region.

Usage

galaxy
Format

A data frame with X rows and Y variables.

Source


Examples

data(galaxy)

plot.AM_mcmc_output

Description

plot some useful informations about the mcmc results

Usage

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

Arguments

x a AM_mcmc_output object
...
all additionnal parameters passed to image command.

said

Usage frequency of the word said in the Brown corpus.

Description

Usage frequency of the word said in the Brown corpus.

Usage

said

Format

A list with 500 observations on the frequency of said in different texts.
summary.AM_mcmc_output

Source
https://www.kaggle.com/nltkdata/brown-corpus

References
pany, Boston.

Examples
data(said)

summary.AM_mcmc_output

summary AM_mcmc_output  Print some useful informations about the
mcmc results

Description
summary AM_mcmc_output
Print some useful informations about the mcmc results

Usage
## S3 method for class  'AM_mcmc_output'
summary(object, ...)

Arguments

object         a AM_mcmc_output object
... all additionnal parameters are ignored
Index

*Topic clusters
  AM_prior_K_Pois, 21

*Topic cluster
  AM_compute_stirling_ricor_abs, 5
  AM_compute_stirling_ricor_log, 6
  AM_find_gamma_Delta, 7
  AM_find_gamma_NegBin, 8
  AM_find_gamma_Pois, 9
  AM_prior_K_Delta, 19
  AM_prior_K_NegBin, 20
  AM_VnkDelta, 25
  AM_VnkNegBin, 26
  AM_VnkPoisson, 27

*Topic datasets
  brain, 29
  carcinoma, 30
  galaxy, 30
  said, 31

*Topic number
  AM_compute_stirling_ricor_abs, 5
  AM_compute_stirling_ricor_log, 6
  AM_find_gamma_Delta, 7
  AM_find_gamma_NegBin, 8
  AM_find_gamma_Pois, 9
  AM_prior_K_Delta, 19
  AM_prior_K_NegBin, 20
  AM_VnkDelta, 25
  AM_VnkNegBin, 26
  AM_VnkPoisson, 27

*Topic prior
  AM_mix_components_prior_dirac, 13
  AM_mix_components_prior_negbin, 13
  AM_mix_components_prior_pois, 14
  AM_mix_weights_prior_gamma, 16
  AM_prior_K_Delta, 19
  AM_prior_K_NegBin, 20
  AM_VnkDelta, 25
  AM_VnkNegBin, 26
  AM_VnkPoisson, 27

AM_clustering_estimation_average, 3
AM_clustering_estimation_laugreen, 3
AM_clustering_estimation_squared_loss, 4
AM_coclustering, 3, 4, 18
AM_coclustering_slow, 5
AM_compute_stirling_ricor_abs, 5
AM_compute_stirling_ricor_log, 6
AM_emp_bayes_uninorm, 6
AM_find_gamma_Delta, 7
AM_find_gamma_NegBin, 8
AM_find_gamma_Pois, 9
AM_mcmc_fit, 10, 11, 13–16, 28
AM_mcmc_output, 3–5, 11, 11, 32
AM_mcmc_parameters, 12
AM_mix_components_prior_dirac, 13
AM_mix_components_prior_negbin, 13
AM_mix_components_prior_pois, 14
AM_mix_weights_prior_gamma, 16
AM_multiber_mix_hyperparams, 17
INDEX

AM_multinorm_mix_hyperparams, 17
AM_plot_similarity_matrix, 18
AM_prior_K_Delta, 19
AM_prior_K_NegBin, 20
AM_prior_K_Pois, 21
AM_unibin_mix_hyperparams, 22
AM_uninorm_mix_hyperparams, 23
AM_unipois_mix_hyperparams, 24
AM_VnkDelta, 25
AM_VnkNegBin, 26
AM_VnkPoisson, 27
AntMAN, 27
AntMAN-package (AntMAN), 27

brain, 29

carcinoma, 30

galaxy, 30

plot.AM_mcmc_output, 31

said, 31

summary.AM_mcmc_output, 32