Package 'Ultimixt'

March 9, 2017

Type Package

Title Bayesian Analysis of Location-Scale Mixture Models using a Weakly Informative Prior

Version 2.1

Date 2017-03-07

Author Kaniav Kamary, Kate Lee

Maintainer Kaniav Kamary <kamary@ceremade.dauphine.fr>

Depends coda, gtools, graphics, grDevices, stats

Description
A generic reference Bayesian analysis of unidimensional mixture distributions obtained by a location-scale parameterisation of the model is implemented. The including functions simulate and summarize posterior samples for location-scale mixture models using a weakly informative prior. There is no need to define priors for scale-location parameters except two hyperparameters in which are associated with a Dirichlet prior for weights and a simplex.

NeedsCompilation no

Repository CRAN

Date/Publication 2017-03-09 00:33:27

R topics documented:

Ultimixt-package *set of R functions for estimating the parameters of mixture distribution with a Bayesian non-informative prior*

Description

Despite a comprehensive literature on estimating mixtures of Gaussian distributions, there does not exist a well-accepted reference Bayesian approach to such models. One reason for the difficulty is the general prohibition against using improper priors (Fruhwirth-Schnatter, 2006) due to the ill-posed nature of such statistical objects. Kamary, Lee and Robert (2017) took advantage of a mean-variance reparametrisation of a Gaussian mixture model to propose improper but valid reference priors in this setting. This R package implements the proposal and computes posterior estimates of the parameters of a Gaussian mixture distribution. The approach applies with an arbitrary number of components. The Ultimixt R package contains an MCMC algorithm function and further functions for summarizing and plotting posterior estimates of the model parameters for any number of components.

Details

Package:	Ultimixt
Type:	Package
Version:	2.1
Date:	2017-03-07
License:	GPL (>=2.0)

Beyond simulating MCMC samples from the posterior distribution of the Gaussian mixture model, this package also produces summaries of the MCMC outputs through numerical and graphical methods.

Note: The proposed parameterisation of the Gaussian mixture distribution is given by

$$f(x|\mu, \sigma, \mathbf{p}, \varphi, \varpi, \xi) = \sum_{i=1}^{k} p_i f\left(x|\mu + \sigma\gamma_i/\sqrt{p_i}, \sigma\eta_i/\sqrt{p_i}\right)$$

under the non-informative prior $\pi(\mu, \sigma) = 1/\sigma$. Here, the vector of the $\gamma_i = \varphi\Psi_i\Big(\varpi, \mathbf{p}\Big)_i$'s belongs to an hypersphere of radius φ intersecting with an hyperplane. It is thus expressed in terms of spherical coordinates within that hyperplane that depend on $k-2$ angular coordinates ϖ_i. Similarly, the vector of $\eta_i = \sqrt{1-\varphi^2}\Psi_i\Big(\xi\Big)_i$'s can be turned into a spherical coordinate in a k-dimensional Euclidean space, involving a radial coordinate $\sqrt{1-\varphi^2}$ and $k-1$ angular coordinates ξ_i. A natural prior for ϖ is made of uniforms, $\varpi_1, \ldots, \varpi_{k-3} \sim U[0, \pi]$ and $\varpi_{k-2} \sim U[0, 2\pi]$, and for φ, we consider a beta prior $Beta(\alpha, \alpha)$. A reference prior on the angles ξ is $(\xi_1, \ldots, \xi_{k-1}) \sim U[0, \pi/2]^{k-1}$ and a Dirichlet prior $Dir(\alpha_0, \ldots, \alpha_0)$ is assigned to the weights $p_1, \ldots, p_k$.

For a Poisson mixture, we consider

$$f(x|\lambda_1,\ldots,\lambda_k) = \frac{1}{x!}\sum_{i=1}^{k} p_i \lambda_i^x e^{-\lambda_i}$$

with a reparameterisation as $\lambda = \mathbf{E}[\mathbf{X}]$ and $\lambda_i = \lambda\gamma_i/p_i$. In this case, we can use the equivalent to the Jeffreys prior for the Poisson distribution, namely, $\pi(\lambda) = 1/\lambda$, since it leads to a well-defined posterior with a single positive observation.

Author(s)

Kaniav Kamary

Maintainer: `<kamary@ceremade.dauphine.fr>`

References

Fruhwirth-Schnatter, S. (2006). Finite Mixture and Markov Switching Models. Springer-Verlag, New York, New York.

Kamary, K., Lee, J.Y., and Robert, C.P. (2017) Weakly informative reparameterisation for location-scale mixtures. arXiv.

See Also

Ultimixt

Examples

```
#K.MixReparametrized(faithful[,2], k=2, alpha0=.5, alpha=.5, Nsim=10000)
```

`K.MixPois` *Sample from a Poisson mixture posterior associated with a noninformative prior and obtained by Metropolis-within-Gibbs sampling*

Description

After having reparameterized the Poisson mixture based on the global mean of the mixture distribution (Kamary et al. (2017)), a Jeffreys prior can be used since it leads a well-defined posterior with a single positive observation. This function returns a sample from the posterior distribution of the parameters of the Poisson mixture. To do so, a Metropolis-within-Gibbs algorithm is applied with an adaptive calibration of the proposal distribution scales. Adaptation is driven by the formally optimal acceptance rates of 0.44 and 0.234 in one and larger dimensions, respectively (Roberts et al.,1997). This algorithm monitors the convergence of the MCMC sequences via Gelman's and Rubin's (1992) criterion.

Usage

```
K.MixPois(xobs, k, alpha0, alpha, Nsim)
```

Arguments

`xobs`	vector of the observations or dataset
`k`	number of components in the mixture model
`alpha0`	hyperparameter of Dirichlet prior distribution of the mixture model weights which is .5 by default
`alpha`	hyperparameter of beta prior distribution of the component mean hyperparameter (noted by γ_i. See Kamary et al. (2017)) which is .5 by default
`Nsim`	number of MCMC iterations after calibration step of proposal scales

Details

The output of this function contains a simulated sample for each parameter of the mixture distribution, the evolution of the proposal scales and acceptance rates over the number of iterations during the calibration stage, and their final values after calibration.

Value

The output of this function contains a list of the following variables, where the dimension of the vectors is the number of simulations:

`mean global`	vector of simulated draws from the conditional posterior of the mixture model mean
`weights`	matrix of simulated draws from the conditional posterior of the mixture model weights with a number of columns equal to the number of components k
`gammas`	matrix of simulated draws from the conditional posterior of the component mean hyperparameters
`accept rat`	vector of resulting acceptance rates of the proposal distributions without calibration step of the proposal scales
`optimal para`	vector of resulting proposal scales after optimisation obtained by adaptive MCMC
`adapt rat`	list of acceptance rates of batch of 50 iterations obtained when calibrating the proposal scales by adaptive MCMC. The number of columns depends on the number of proposal distributions.
`adapt scale`	list of proposal scales calibrated by adaptive MCMC for each batch of 50 iterations with respect to the optimal acceptance rate. The number of columns depends on the number of proposal distribution scales.
`component means`	matrix of MCMC samples of the component means of the mixture model with a number of columns equal to k

Note

If the number of MCMC iterations specified in the input of this function exceeds 15,000, after each 1000 supplementry iterations the convergence of simulated chains is checked using the convergence monitoring technique by Gelman and Rubin (1992).

Author(s)

Kaniav Kamary

References

Kamary, K., Lee, J.Y., and Robert, C.P. (2017) Weakly informative reparameterisation of location-scale mixtures. arXiv.

Robert, C. and Casella, G. (2009). Introducing Monte Carlo Methods with R. Springer-Verlag.

Roberts, G. O., Gelman, A. and Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. Ann. Applied Probability, 7, 110–120.

Gelman, A. and Rubin, D. (1992). Inference from iterative simulation using multiple sequences (with discussion). Statistical Science, 457–472.

See Also

Ultimixt

Examples

```
#N=500
#U =runif(N)
#xobs = rep(NA,N)
#for(i in 1:N){
#    if(U[i]<.6){
#        xobs[i] = rpois(1,lambda=1)
#    }else{
#        xobs[i] = rpois(1,lambda=5)
#    }
#}
#estimate=K.MixPois(xobs, k=2, alpha0=.5, alpha=.5, Nsim=10000)
```

K.MixReparametrized *Sample from a Gaussian mixture posterior associated with a noninformative prior and obtained by Metropolis-within-Gibbs sampling*

Description

This function returns a sample simulated from the posterior distribution of the parameters of a Gaussian mixture under a non-informative prior. This prior is derived from a mean-variance reparameterisation of the mixture distribution, as proposed by Kamary et al. (2017). The algorithm is a Metropolis-within-Gibbs scheme with an adaptive calibration of the proposal distribution scales. Adaptation is driven by the formally optimal acceptance rates of 0.44 and 0.234 in one and larger dimensions, respectively (Roberts et al.,1997). This algorithm monitors the convergence of the MCMC sequences via Gelman's and Rubin's (1992) criterion.

Usage

```
K.MixReparametrized(xobs, k, alpha0, alpha, Nsim)
```

Arguments

- `xobs` vector of the observations or dataset
- `k` number of components in the mixture model
- `alpha0` hyperparameter of Dirichlet prior distribution of the mixture model weights which is .5 by default
- `alpha` hyperparameter of beta prior distribution of the radial coordinate which is .5 by default
- `Nsim` number of MCMC iterations after calibration step of proposal scales

Details

The output of this function contains a simulated sample for each parameter of the mixture distribution, the evolution of the proposal scales and acceptance rates over the number of iterations during the calibration stage, and their final values after calibration.

Value

The output of this function is a list of the following variables, where the dimension of the vectors is the number of simulations:

- `mean global` vector of simulated draws from the conditional posterior of the mixture model mean
- `sigma global` vector of simulated draws from the conditional posterior of the mixture model standard deviation
- `weights` matrix of simulated draws from the conditional posterior of the mixture model weights with a number of columns equal to the number of components k
- `angles xi` matrix of simulated draws from the conditional posterior of the angular coordinates of the component standard deviations with a number of columns equal to $k-1$
- `phi` vector of simulated draws from the conditional posterior of the radian coordinate
- `angles varpi` matrix of simulated draws from the conditional posterior of the angular coordinates of the component means with a number of columns equal to $k-2$
- `accept rat` vector of resulting acceptance rates of the proposal distributions without calibration step of the proposal scales
- `optimal para` vector of resulting proposal scales after optimisation obtained by adaptive MCMC
- `adapt rat` list of acceptance rates of batch of 50 iterations obtained when calibrating the proposal scales by adaptive MCMC. The number of columns depends on the number of proposal distributions.
- `adapt scale` list of proposal scales calibrated by adaptive MCMC for each batch of 50 iterations with respect to the optimal acceptance rate. The number of columns depends on the number of proposal distribution scales.
- `component means` matrix of MCMC samples of the component means of the mixture model with a number of columns equal to k

`component sigmas`
: matrix of MCMC samples of the component standard deviations of the mixture model with a number of columns equal to k

Note

If the number of MCMC iterations specified in the input of this function exceeds 15,000, after each 1000 supplementry iterations the convergence of simulated chains is checked using the convergence monitoring technique by Gelman and Rubin (1992).

Author(s)

Kaniav Kamary

References

Kamary, K., Lee, J.Y., and Robert, C.P. (2017) Weakly informative reparameterisation of location-scale mixtures. arXiv.

Robert, C. and Casella, G. (2009). Introducing Monte Carlo Methods with R. Springer-Verlag.

Roberts, G. O., Gelman, A. and Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. Ann. Applied Probability, 7, 110–120.

Gelman, A. and Rubin, D. (1992). Inference from iterative simulation using multiple sequences (with discussion). Statistical Science, 457–472.

See Also

Ultimixt

Examples

```
#data(faithful)
#xobs=faithful[,1]
#estimate=K.MixReparametrized(xobs, k=2, alpha0=.5, alpha=.5, Nsim=10000)
```

Plot.MixReparametrized

plot of the MCMC output produced by K.MixReparametrized

Description

This is a generic function for a graphical rendering of the MCMC samples produced by K.MixReparametrized function. The function draws boxplots for unimodal variables and for multimodal arguments after clustering them by applying a k-means algorithm. It also plots line charts for other variables.

Usage

```
Plot.MixReparametrized(xobs, estimate)
```

Arguments

`xobs`	vector of the observations
`estimate`	output of the K. MixReparametrized function

Details

Boxplots are produced using the boxplot.default method.

Value

The output of this function consists of

`boxplot`	three boxplots for the radial coordinates, the mean and the standard deviation of the mixture distribution, k boxplots for each of the mixture model weights, component means and component standard deviations.
`histogram`	an histogram of the observations against an overlaid curve of the density estimate, obtained by averaging over all mixtures corresponding to the MCMC draws,
`line chart`	line charts that report the evolution of the proposal scales and of the acceptance rates over the number of batch of 50 iterations.

Note

The mixture density estimate is based on the draws simulated of the parameters obtained by K.MixReparametrized function.

Author(s)

Kaniav Kamary

References

Kamary, K., Lee, J.Y., and Robert, C.P. (2017) Weakly informative reparameterisation of location-scale mixtures. arXiv.

See Also

`K.MixReparametrized`

Examples

```
#data(faithful)
#xobs=faithful[,1]
#estimate=K.MixReparametrized(xobs, k=2, alpha0=.5, alpha=.5, Nsim=20000)
#plo=Plot.MixReparametrized(xobs, estimate)
```

SM.MAP.MixReparametrized

summary of the output produced by K.MixReparametrized

Description

Label switching in a simulated Markov chain produced by K.MixReparametrized is removed by the technique of Marin et al. (2004). Namely, component labels are reorded by the shortest Euclidian distance between a posterior sample and the maximum a posteriori (MAP) estimate. Let θ_i be the i-th vector of computed component means, standard deviations and weights. The MAP estimate is derived from the MCMC sequence and denoted by θ_{MAP}. For a permutation $\tau \in \Im_k$ the labelling of θ_i is reordered by

$$\tilde{\theta}_i = \tau_i(\theta_i)$$

where $\tau_i = \arg\min_{\tau \in \Im_k} || \tau(\theta_i) - \theta_{MAP} ||$.

Angular parameters $\xi_1^{(i)}, \ldots, \xi_{k-1}^{(i)}$ and $\varpi_1^{(i)}, \ldots, \varpi_{k-2}^{(i)}$s are derived from $\tilde{\theta}_i$. There exists an unique solution in $\varpi_1^{(i)}, \ldots, \varpi_{k-2}^{(i)}$ while there are multiple solutions in $\xi^{(i)}$ due to the symmetry of $| \cos(\xi) |$ and $| \sin(\xi) |$. The output of $\xi_1^{(i)}, \ldots, \xi_{k-1}^{(i)}$ only includes angles on $[-\pi, \pi]$.

The label of components of θ_i (before the above transform) is defined by

$$\tau_i^* = \arg\min_{\tau \in \Im_k} || \theta_i - \tau(\theta_{MAP}) || .$$

The number of label switching occurrences is defined by the number of changes in τ^*.

Usage

```
SM.MAP.MixReparametrized(estimate, xobs, alpha0, alpha)
```

Arguments

`estimate`	Output of K.MixReparametrized
`xobs`	Data set
`alpha0`	Hyperparameter of Dirichlet prior distribution of the mixture model weights
`alpha`	Hyperparameter of beta prior distribution of the radial coordinate

Details

Details.

Value

`MU`	Matrix of MCMC samples of the component means of the mixture model
`SIGMA`	Matrix of MCMC samples of the component standard deviations of the mixture model

`P`	Matrix of MCMC samples of the component weights of the mixture model
`Ang_SIGMA`	Matrix of computed ξ's corresponding to SIGMA
`Ang_MU`	Matrix of computed ϖ's corresponding to MU. This output only appears when $k > 2$.
`Global_Mean`	Mean, median and 95% credible interval for the global mean parameter
`Global_Std`	Mean, median and 95% credible interval for the global standard deviation parameter
`Phi`	Mean, median and 95% credible interval for the radius parameter
`component_mu`	Mean, median and 95% credible interval of MU
`component_sigma`	Mean, median and 95% credible interval of SIGMA
`component_p`	Mean, median and 95% credible interval of P
`l_stay`	Number of MCMC iterations between changes in labelling
`n_switch`	Number of label switching occurrences

Note

Note.

Author(s)

Kate Lee

References

Marin, J.-M., Mengersen, K. and Robert, C. P. (2004) Bayesian Modelling and Inference on Mixtures of Distributions, Handbook of Statistics, Elsevier, Volume 25, Pages 459–507.

See Also

`K.MixReparametrized`

Examples

```
#data(faithful)
#xobs=faithful[,1]
#estimate=K.MixReparametrized(xobs,k=2,alpha0=0.5,alpha=0.5,Nsim=1e4)
#result=SM.MAP.MixReparametrized(estimate,xobs,alpha0=0.5,alpha=0.5)
```

SM.MixPois — *summary of the output produced by K.MixPois*

Description

This generic function summarizes the MCMC samples produced by K.MixPois when several estimation methods have been invoked depending on the unimodality or multimodality of the argument.

Usage

```
SM.MixPois(estimate, xobs)
```

Arguments

`estimate`	output of K.MixPois
`xobs`	vector of observations

Details

The output of this function contains posterior point estimates for all parameters of the reparameterized Poisson mixture model. It summarizes unimodal MCMC samples by computing measures of centrality, including mean and median, while multimodal outputs require a preprocessing, due to the label switching phenomenon (Jasra et al., 2005). The summary measures are then computed after performing a multi-dimensional k-means clustering (Hartigan and Wong, 1979) following the suggestion of Fruhwirth-Schnatter (2006).

Value

`lambda`	vector of mean and median of simulated draws from the conditional posterior of the mixture model mean
`gamma.i`	vector of mean and median of simulated draws from the conditional posterior of the component mean hyperparameters; $i = 1, \ldots, k$
`weight.i`	vector of mean and median of simulated draws from the conditional posterior of the component weights of the mixture distribution; $i = 1, \ldots, k$
`lambda.i`	vector of mean and median of simulated draws from the conditional posterior of the component means of the mixture distribution; $i = 1, \ldots, k$
`Acc rat`	vector of final acceptance rate of the proposal distributions of the algorithm with no calibration stage for the proposal scales
`Opt scale`	vector of optimal proposal scales obtained the by calibration stage

Note

For multimodal outputs such as the mixture model weights, component means, and component mean hyperparameters, for each MCMC draw, first the labels of the weights $p_i, i = 1, \ldots, k$ and corresponding component means are permuted in such a way that $p_1 \leq \ldots \leq p_k$. Then the posterior component means are partitioned into k clusters by applying a standard k-means algorithm with k clusters, following Fruhwirth-Schnatter (2006) method. The obtained classification sequence was then used to reorder and identify the other component-specific parameters, namely component mean hyperparameters and weights. For each group, cluster centers are considered as parameter estimates.

Author(s)

Kaniav Kamary

References

Jasra, A., Holmes, C. and Stephens, D. (2005). Markov Chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. Statistical Science, 20, 50–67.

Hartigan, J. A. and Wong, M. A. (1979). A K-means clustering algorithm. Applied Statistics 28, 100–108.

Fruhwirth-Schnatter, S. (2006). Finite mixture and Markov switching models. Springer-Verlag.

See Also

K.MixPois

Examples

```
N=500
U =runif(N)
xobs = rep(NA,N)
for(i in 1:N){
    if(U[i]<.6){
        xobs[i] = rpois(1,lambda=1)
    }else{
        xobs[i] = rpois(1,lambda=5)
    }
}
#estimate=K.MixPois(xobs, k=2, alpha0=.5, alpha=.5, Nsim=10000)
#SM.MixPois(estimate, xobs)
#plot(estimate[[8]][,1],estimate[[2]][,1],pch=19,col="skyblue",cex=0.5,xlab="lambda",ylab="p")
#points(estimate[[8]][,2], estimate[[2]][,2], pch=19, col="gold", cex=0.5)
#points(c(1,5), c(0.6,0.4), pch=19, cex=1)
```

SM.MixReparametrized *summary of the output produced by K.MixReparametrized*

Description

This is a generic function that summarizes the MCMC samples produced by K.MixReparametrized. The function invokes several estimation methods which choice depends on the unimodality or multimodality of the argument.

Usage

```
SM.MixReparametrized(xobs, estimate)
```

Arguments

`xobs`	vector of observations
`estimate`	output of K.MixReparametrized

Details

This function outputs posterior point estimates for all parameters of the mixture model. They mostly differ from the generaly useless posterior means. The output summarizes unimodal MCMC samples by computing measures of centrality, including mean and median, while multimodal outputs require a pre-processing, due to the label switching phenomenon (Jasra et al., 2005). The summary measures are then computed after performing a multi-dimensional k-means clustering (Hartigan and Wong, 1979) following the suggestion of Fruhwirth-Schnatter (2006).

Value

`Mean`	vector of mean and median of simulated draws from the conditional posterior of the mixture model mean
`Sd`	vector of mean and median of simulated draws from the conditional posterior of the mixture model standard deviation
`Phi`	vector of mean and median of simulated draws from the conditional posterior of the radial coordinate
`Angles. 1.`	vector of means of the angular coordinates used for the component means in the mixture distribution
`Angles. 2.`	vector of means of the angular coordinates used for the component standard deviations in the mixture distribution
`weight.i`	vector of mean and median of simulated draws from the conditional posterior of the component weights of the mixture distribution; $i = 1, \ldots, k$
`mean.i`	vector of mean and median of simulated draws from the conditional posterior of the component means of the mixture distribution; $i = 1, \ldots, k$
`sd.i`	vector of mean and median of simulated draws from the conditional posterior of the component standard deviations of the mixture distribution; $i = 1, \ldots, k$

`Acc rat` vector of final acceptance rate of the proposal distributions of the algorithm with no calibration stage for the proposal scales

`Opt scale` vector of optimal proposal scales obtained the by calibration stage

Note

For multimodal outputs such as the mixture model weights, component means, and component variances, for each MCMC draw, first the labels of the weights $p_i, i = 1, \ldots, k$ and corresponding component means and standard deviations are permuted in such a way that $p_1 \leq \ldots \leq p_k$. Then the component means and standard deviations are jointly partitioned into k clusters by applying a standard k-means algorithm with k clusters, following Fruhwirth-Schnatter (2006) method. The obtained classification sequence was then used to reorder and identify the other component-specific parameters, namely component mean hyperparameters and weights. For each group, cluster centers are considered as parameter estimates.

Author(s)

Kaniav Kamary

References

Jasra, A., Holmes, C. and Stephens, D. (2005). Markov Chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. Statistical Science, 20, 50–67.

Hartigan, J. A. and Wong, M. A. (1979). A K-means clustering algorithm. Applied Statistics 28, 100–108.

Fruhwirth-Schnatter, S. (2006). Finite mixture and Markov switching models. Springer-Verlag.

See Also

K.MixReparametrized

Examples

```
#data(faithful)
#xobs=faithful[,1]
#estimate=K.MixReparametrized(xobs, k=2, alpha0=.5, alpha=.5, Nsim=20000)
#summari=SM.MixReparametrized(xobs,estimate)
```

Index