Package 'AdMit'

April 20, 2020

Version 2.1.5

Date 2020-04-20

Title Adaptive Mixture of Student-t Distributions

Maintainer David Ardia <david.ardia.ch@gmail.com>

Depends mvtnorm

Suggests coda

Description Provides functions to perform the fitting of an adaptive mixture of Student-t distributions to a target density through its kernel function as described in Ardia et al. (2009) <doi:10.18637/jss.v029.i03>. The mixture approximation can then be used as the importance density in importance sampling or as the candidate density in the Metropolis-Hastings algorithm to obtain quantities of interest for the target density itself.

BugReports https://github.com/ArdiaD/AdMit/issues

URL https://github.com/ArdiaD/AdMit

License GPL (>= 2)

RoxygenNote 5.0.1

NeedsCompilation yes

Author David Ardia [aut, cre, cph] (<https://orcid.org/0000-0003-2823-782X>), Lennart Hoogerheide [ctb], Herman van Dijk [ctb]

Repository CRAN

Date/Publication 2020-04-20 21:50:11 UTC

R topics documented:

AdMit *Adaptive Mixture of Student-t Distributions*

Description

Function which performs the fitting of an adaptive mixture of Student-t distributions to approximate a target density through its kernel function

Usage

```
AdMit(KERNEL, mu0, Sigma0 = NULL, control = list(), ...)
```

Arguments

KERNEL	kernel function of the target density on which the adaptive mixture is fitted. This function should be vectorized for speed purposes (i.e., its first argument should be a matrix and its output a vector). Moreover, the function must contain the logical argument log. If log = TRUE, the function returns (natural) logarithm values of the kernel function. NA and NaN values are not allowed. (See *Details* for examples of KERNEL implementation.)
mu0	initial value in the first stage optimization (for the location of the first Student-t component) in the adaptive mixture, or location of the first Student-t component if Sigma0 is not NULL.
Sigma0	scale matrix of the first Student-t component (square, symmetric and positive definite). Default: Sigma0 = NULL, i.e., the scale matrix of the first Student-t component is estimated by the function AdMit.
control	control parameters (see *Details*).
...	further arguments to be passed to KERNEL.

Details

The argument KERNEL is the kernel function of the target density, and it should be vectorized for speed purposes.

As a first example, consider the kernel function proposed by Gelman-Meng (1991):

$$k(x_1, x_2) = \exp\left(-\frac{1}{2}[Ax_1^2x_2^2 + x_1^2 + x_2^2 - 2Bx_1x_2 - 2C_1x_1 - 2C_2x_2]\right)$$

where commonly used values are $A = 1$, $B = 0$, $C_1 = 3$ and $C_2 = 3$.

A vectorized implementation of this function might be:

```
GelmanMeng <- function(x, A = 1, B = 0, C1 = 3, C2 = 3, log = TRUE)
{
  if (is.vector(x))
    x <- matrix(x, nrow = 1)
  r <- -.5 * (A * x[,1]^2 * x[,2]^2 + x[,1]^2 + x[,2]^2
```

```
            - 2 * B * x[,1] * x[,2] - 2 * C1 * x[,1] - 2 * C2 * x[,2])
    if (!log)
      r <- exp(r)
    as.vector(r)
  }
```

This way, we may supply a point (x_1, x_2) for `x` and the function will output a single value (i.e., the kernel estimated at this point). But the function is vectorized, in the sense that we may supply a $(N \times 2)$ matrix of values for `x`, where rows of `x` are points (x_1, x_2) and the output will be a vector of length N, containing the kernel values for these points. Since the `AdMit` procedure evaluates `KERNEL` for a large number of points, a vectorized implementation is important. Note also the additional argument `log = TRUE` which is used for numerical stability.

As a second example, consider the following (simple) econometric model:

$$y_t \sim \ i.i.d.\ N(\mu, \sigma^2) \quad t = 1, \ldots, T$$

where μ is the mean value and σ is the standard deviation. Our purpose is to estimate $\theta = (\mu, \sigma)$ within a Bayesian framework, based on a vector y of T observations; the kernel thus consists of the product of the prior and the likelihood function. As previously mentioned, the kernel function should be vectorized, i.e., treat a $(N \times 2)$ matrix of points θ for which the kernel will be evaluated. Using the common (Jeffreys) prior $p(\theta) = \frac{1}{\sigma}$ for $\sigma > 0$, a vectorized implementation of the kernel function might be:

```
  KERNEL <- function(theta, y, log = TRUE)
  {
    if (is.vector(theta))
      theta <- matrix(theta, nrow = 1)

    ## sub function which returns the log-kernel for a given
    ## thetai value (i.e., a given row of theta)
    KERNEL_sub <- function(thetai)
    {
      if (thetai[2] > 0) ## check if sigma>0
{ ## if yes, compute the log-kernel at thetai
        r <- - log(thetai[2])
      + sum(dnorm(y, thetai[1], thetai[2], TRUE))
}
else
{ ## if no, returns -Infinity
  r <- -Inf
}
as.numeric(r)
    }

    ## 'apply' on the rows of theta (faster than a for loop)
    r <- apply(theta, 1, KERNEL_sub)
    if (!log)
      r <- exp(r)
```

```
  as.numeric(r)
}
```

Since this kernel function also depends on the vector y, it must be passed to `KERNEL` in the `AdMit` function. This is achieved via the argument ..., i.e., `AdMit(KERNEL,mu = c(0,1),y = y)`.

To gain even more speed, implementation of `KERNEL` might rely on C or Fortran code using the functions `.C` and `.Fortran`. An example is provided in the file '`AdMitJSS.R`' in the package's folder.

The argument `control` is a list that can supply any of the following components:

`Ns` number of draws used in the evaluation of the importance sampling weights (integer number not smaller than 100). Default: `Ns = 1e5`.

`Np` number of draws used in the optimization of the mixing probabilities (integer number not smaller than 100 and not larger than `Ns`). Default: `Np = 1e3`.

`Hmax` maximum number of Student-t components in the adaptive mixture (integer number not smaller than one). Default: `Hmax = 10`.

`df` degrees of freedom parameter of the Student-t components (real number not smaller than one). Default: `df = 1`.

`CVtol` tolerance for the relative change of the coefficient of variation (real number in [0,1]). The adaptive algorithm stops if the new component leads to a relative change in the coefficient of variation that is smaller or equal than `CVtol`. Default: `CVtol = 0.1`, i.e., 10%.

`weightNC` weight assigned to the new Student-t component of the adaptive mixture as a starting value in the optimization of the mixing probabilities (real number in [0,1]). Default: `weightNC = 0.1`, i.e., 10%.

`trace` tracing information on the adaptive fitting procedure (logical). Default: `trace = FALSE`, i.e., no tracing information.

`IS` should importance sampling be used to estimate the mode and the scale matrix of the Student-t components (logical). Default: `IS = FALSE`, i.e., use numerical optimization instead.

`ISpercent` vector of percentage(s) of largest weights used to estimate the mode and the scale matrix of the Student-t components of the adaptive mixture by importance sampling (real number(s) in [0,1]). Default: `ISpercent = c(0.05,0.15,0.3)`, i.e., 5%, 15% and 30%.

`ISscale` vector of scaling factor(s) used to rescale the scale matrix of the mixture components (real positive numbers). Default: `ISscale = c(1,0.25,4)`.

`trace.mu` Tracing information on the progress in the optimization of the mode of the mixture components (non-negative integer number). Higher values may produce more tracing information (see the source code of the function `optim` for further details). Default: `trace.mu = 0`, i.e., no tracing information.

`maxit.mu` maximum number of iterations used in the optimization of the modes of the mixture components (positive integer). Default: `maxit.mu = 500`.

`reltol.mu` relative convergence tolerance used in the optimization of the modes of the mixture components (real number in [0,1]). Default: `reltol.mu = 1e-8`.

`trace.p`, `maxit.p`, `reltol.p` the same as for the arguments above, but for the optimization of the mixing probabilities of the mixture components.

Value

A list with the following components:

CV: vector (of length H) of coefficients of variation of the importance sampling weights.

mit: list (of length 4) containing information on the fitted mixture of Student-t distributions, with the following components:

p: vector (of length H) of mixing probabilities. mu: matrix (of size $H \times d$) containing the vectors of modes (in row) of the mixture components. Sigma: matrix (of size $H \times d^2$) containing the scale matrices (in row) of the mixture components. df: degrees of freedom parameter of the Student-t components.

where $H(\geq 1)$ is the number of components in the adaptive mixture of Student-t distributions and $d(\geq 1)$ is the dimension of the first argument in KERNEL.

summary: data frame containing information on the optimization procedures. It returns for each component of the adaptive mixture of Student-t distribution: 1. the method used to estimate the mode and the scale matrix of the Student-t component ('USER' if Sigma0 is provided by the user; numerical optimization: 'BFGS', 'Nelder-Mead'; importance sampling: 'IS', with percentage(s) of importance weights used and scaling factor(s)); 2. the time required for this optimization; 3. the method used to estimate the mixing probabilities ('NLMINB', 'BFGS', 'Nelder-Mead', 'NONE'); 4. the time required for this optimization; 5. the coefficient of variation of the importance sampling weights.

Note

By using AdMit you agree to the following rules:

- You must cite Ardia et al. (2009a,b) in working papers and published papers that use AdMit. Use citation("AdMit").
- You must place the following URL in a footnote to help others find AdMit: https://CRAN.R-project.org/package=AdMit.
- You assume all risk for the use of AdMit.

Further details and examples of the R package AdMit can be found in Ardia et al. (2009a,b). See also the package vignette by typing vignette("AdMit").

Further details on the core algorithm are given in Hoogerheide (2006), Hoogerheide, Kaashoek, van Dijk (2007) and Hoogerheide, van Dijk (2008).

The adaptive mixture mit returned by the function AdMit is used by the function AdMitIS to perform importance sampling using mit as the importance density or by the function AdMitMH to perform independence chain Metropolis-Hastings sampling using mit as the candidate density.

Please cite the package in publications. Use citation("AdMit").

Author(s)

David Ardia for the R port, Lennart F. Hoogerheide and Herman K. van Dijk for the AdMit algorithm.

References

Ardia, D., Hoogerheide, L.F., van Dijk, H.K. (2009a). AdMit: Adaptive Mixture of Student-t Distributions. *The R Journal* **1**(1), pp.25-30. https://journal.R-project.org/archive/2009-1/

Ardia, D., Hoogerheide, L.F., van Dijk, H.K. (2009b). Adaptive Mixture of Student-t Distributions as a Flexible Candidate Distribution for Efficient Simulation: The R Package AdMit. *Journal of Statistical Software* **29**(3), pp.1-32. doi: 10.18637/jss.v029.i03

Gelman, A., Meng, X.-L. (1991). A Note on Bivariate Distributions That Are Conditionally Normal. *The American Statistician* **45**(2), pp.125-126.

Hoogerheide, L.F. (2006). *Essays on Neural Network Sampling Methods and Instrumental Variables*. PhD thesis, Tinbergen Institute, Erasmus University Rotterdam (NL). ISBN: 9051708261. (Book nr. 379 of the Tinbergen Institute Research Series.)

Hoogerheide, L.F., Kaashoek, J.F., van Dijk, H.K. (2007). On the Shape of Posterior Densities and Credible Sets in Instrumental Variable Regression Models with Reduced Rank: An Application of Flexible Sampling Methods using Neural Networks. *Journal of Econometrics* **139**(1), pp.154-180. doi: 10.1016/j.jeconom.2006.06.009

Hoogerheide, L.F., van Dijk, H.K. (2008). Possibly Ill-Behaved Posteriors in Econometric Models: On the Connection between Model Structures, Non-elliptical Credible Sets and Neural Network Simulation Techniques. *Tinbergen Institute discussion paper* **2008-036/4**.

See Also

AdMitIS for importance sampling using an adaptive mixture of Student-t distributions as the importance density, AdMitMH for the independence chain Metropolis-Hastings algorithm using an adaptive mixture of Student-t distributions as the candidate density.

Examples

```
## NB : Low number of draws for speedup. Consider using more draws!
## Gelman and Meng (1991) kernel function
GelmanMeng <- function(x, A = 1, B = 0, C1 = 3, C2 = 3, log = TRUE)
{
  if (is.vector(x))
    x <- matrix(x, nrow = 1)
  r <- -.5 * (A * x[,1]^2 * x[,2]^2 + x[,1]^2 + x[,2]^2
            - 2 * B * x[,1] * x[,2] - 2 * C1 * x[,1] - 2 * C2 * x[,2])
  if (!log)
    r <- exp(r)
  as.vector(r)
}

## Run AdMit (with default values)
set.seed(1234)
```

```
outAdMit <- AdMit(GelmanMeng, mu0 = c(0.0, 0.1), control = list(Ns = 1e4))
print(outAdMit)

## Run AdMit (using importance sampling to estimate
## the modes and the scale matrices)
set.seed(1234)
outAdMit <- AdMit(KERNEL = GelmanMeng,
                  mu0 = c(0.0, 0.1),
                  control = list(IS = TRUE, Ns = 1e4))
print(outAdMit)
```

AdMitIS *Importance Sampling using an Adaptive Mixture of Student-t Distributions as the Importance Density*

Description

Performs importance sampling using an adaptive mixture of Student-t distributions as the importance density

Usage

```
AdMitIS(N = 1e5, KERNEL, G = function(theta){theta}, mit = list(), ...)
```

Arguments

N	number of draws used in importance sampling (positive integer number). Default: N = 1e5.
KERNEL	kernel function of the target density on which the adaptive mixture of Student-t distributions is fitted. This function should be vectorized for speed purposes (i.e., its first argument should be a matrix and its output a vector). Moreover, the function must contain the logical argument log. If log = TRUE, the function returns (natural) logarithm values of the kernel function. NA and NaN values are not allowed. (See the function AdMit for examples of KERNEL implementation.)
G	function of interest used in importance sampling (see *Details*).
mit	list containing information on the mixture approximation (see *Details*).
...	further arguments to be passed to KERNEL and/or G.

Details

The AdMitIS function estimates $E_p[g(\theta)]$, where p is the target density, g is an (integrable w.r.t. p) function and E denotes the expectation operator, by importance sampling using an adaptive mixture of Student-t distributions as the importance density.

By default, the function G is given by:

```
G <- function(theta)
{
  theta
}
```

and therefore, AdMitIS estimates the mean of theta by importance sampling. For other definitions of G, see *Examples*.

The argument mit is a list containing information on the mixture approximation. The following components must be provided:

p vector (of length H) of mixing probabilities.

mu matrix (of size $H \times d$) containing the vectors of modes (in row) of the mixture components.

Sigma matrix (of size $H \times d^2$) containing the scale matrices (in row) of the mixture components.

df degrees of freedom parameter of the Student-t components (real number not smaller than one).

where $H(\geq 1)$ is the number of components of the adaptive mixture of Student-t distributions and $d(\geq 1)$ is the dimension of the first argument in KERNEL. Typically, mit is estimated by the function AdMit.

Value

A list with the following components:

ghat: a vector containing the importance sampling estimates. NSE: a vector containing the numerical standard error of the components of ghat. RNE: a vector containing the relative numerical efficiency of the components of ghat.

Note

Further details and examples of the R package AdMit can be found in Ardia, Hoogerheide, van Dijk (2009a,b). See also the package vignette by typing vignette("AdMit").

Further information on importance sampling can be found in Geweke (1989) or Koop (2003).

Please cite the package in publications. Use citation("AdMit").

Author(s)

David Ardia

References

Ardia, D., Hoogerheide, L.F., van Dijk, H.K. (2009a). AdMit: Adaptive Mixture of Student-t Distributions. *The R Journal* **1**(1), pp.25-30. https://journal.R-project.org/archive/2009-1/

Ardia, D., Hoogerheide, L.F., van Dijk, H.K. (2009b). Adaptive Mixture of Student-t Distributions as a Flexible Candidate Distribution for Efficient Simulation: The R Package AdMit. *Journal of Statistical Software* **29**(3), pp.1-32. doi: 10.18637/jss.v029.i03

Geweke, J.F. (1989). Bayesian Inference in Econometric Models Using Monte Carlo Integration. *Econometrica* **57**(6), pp.1317-1339.

Koop, G. (2003). *Bayesian Econometrics*. Wiley-Interscience (London, UK). ISBN: 0470845678.

See Also

AdMit for fitting an adaptive mixture of Student-t distributions to a target density through its KERNEL function, AdMitMH for the independence chain Metropolis-Hastings algorithm using an adaptive mixture of Student-t distributions as the candidate density.

Examples

```
## NB : Low number of draws for speedup. Consider using more draws!
## Gelman and Meng (1991) kernel function
GelmanMeng <- function(x, A = 1, B = 0, C1 = 3, C2 = 3, log = TRUE)
{
  if (is.vector(x))
    x <- matrix(x, nrow = 1)
  r <- -.5 * (A * x[,1]^2 * x[,2]^2 + x[,1]^2 + x[,2]^2
            - 2 * B * x[,1] * x[,2] - 2 * C1 * x[,1] - 2 * C2 * x[,2])
  if (!log)
    r <- exp(r)
  as.vector(r)
}

## Run the AdMit function to fit the mixture approximation
set.seed(1234)
outAdMit <- AdMit(KERNEL = GelmanMeng,
                  mu0 = c(0.0, 0.1), control = list(Ns = 1e4))

## Use importance sampling with the mixture approximation as the
## importance density
outAdMitIS <- AdMitIS(N = 1e4, KERNEL = GelmanMeng, mit = outAdMit$mit)
print(outAdMitIS)
```

AdMitMH *Independence Chain Metropolis-Hastings Algorithm using an Adaptive Mixture of Student-t Distributions as the Candidate Density*

Description

Performs independence chain Metropolis-Hastings (M-H) sampling using an adaptive mixture of Student-t distributions as the candidate density

Usage

```
AdMitMH(N = 1e5, KERNEL, mit = list(), ...)
```

Arguments

N	number of draws generated by the independence chain M-H algorithm (positive integer number). Default: N = 1e5.
KERNEL	kernel function of the target density on which the adaptive mixture is fitted. This function should be vectorized for speed purposes (i.e., its first argument should be a matrix and its output a vector). Moreover, the function must contain the logical argument log. If log = TRUE, the function returns (natural) logarithm values of kernel function. NA and NaN values are not allowed. (See the function AdMit for examples of KERNEL implementation.)
mit	list containing information on the mixture approximation (see *Details*).
...	further arguments to be passed to KERNEL.

Details

The argument mit is a list containing information on the adaptive mixture of Student-t distributions. The following components must be provided:

p vector (of length H) of mixing probabilities.

mu matrix (of size $H \times d$) containing the vectors of modes (in row) of the mixture components.

Sigma matrix (of size $H \times d^2$) containing the scale matrices (in row) of the mixture components.

df degrees of freedom parameter of the Student-t components (real number not smaller than one).

where $H(\geq 1)$ is the number of components and $d(\geq 1)$ is the dimension of the first argument in KERNEL. Typically, mit is estimated by the function AdMit.

Value

A list with the following components:

draws: matrix (of size N$\times d$) of draws generated by the independence chain M-H algorithm, where N (≥ 1) is the number of draws and $d(\geq 1)$ is the dimension of the first argument in KERNEL.

accept: acceptance rate of the independence chain M-H algorithm.

Note

Further details and examples of the R package AdMit can be found in Ardia, Hoogerheide and van Dijk (2009a,b). See also the package vignette by typing vignette("AdMit").

Further information on the Metropolis-Hastings algorithm can be found in Chib and Greenberg (1995) and Koop (2003).

Please cite the package in publications. Use citation("AdMit").

Author(s)

David Ardia

References

Ardia, D., Hoogerheide, L.F., van Dijk, H.K. (2009a). AdMit: Adaptive Mixture of Student-t Distributions. *The R Journal* **1**(1), pp.25-30. https://journal.R-project.org/archive/2009-1/

Ardia, D., Hoogerheide, L.F., van Dijk, H.K. (2009b). Adaptive Mixture of Student-t Distributions as a Flexible Candidate Distribution for Efficient Simulation: The R Package AdMit. *Journal of Statistical Software* **29**(3), pp.1-32. doi: 10.18637/jss.v029.i03

Chib, S., Greenberg, E. (1995). Understanding the Metropolis-Hasting Algorithm. *The American Statistician* **49**(4), pp.327-335.

Koop, G. (2003). *Bayesian Econometrics*. Wiley-Interscience (London, UK). ISBN: 0470845678.

See Also

AdMitIS for importance sampling using an adaptive mixture of Student-t distributions as the importance density, AdMit for fitting an adaptive mixture of Student-t distributions to a target density through its KERNEL function; the package coda for MCMC output analysis.

Examples

```
## NB : Low number of draws for speedup. Consider using more draws!
## Gelman and Meng (1991) kernel function
GelmanMeng <- function(x, A = 1, B = 0, C1 = 3, C2 = 3, log = TRUE)
{
  if (is.vector(x))
    x <- matrix(x, nrow = 1)
    r <- -.5 * (A * x[,1]^2 * x[,2]^2 + x[,1]^2 + x[,2]^2
              - 2 * B * x[,1] * x[,2] - 2 * C1 * x[,1] - 2 * C2 * x[,2])
    if (!log)
      r <- exp(r)
    as.vector(r)
  }

## Run the AdMit function to fit the mixture approximation
set.seed(1234)
outAdMit <- AdMit(KERNEL = GelmanMeng,
                  mu0 = c(0.0, 0.1), control = list(Ns = 1e4))

## Run M-H using the mixture approximation as the candidate density
outAdMitMH <- AdMitMH(N = 1e4, KERNEL = GelmanMeng, mit = outAdMit$mit)
options(digits = 4, max.print = 40)
print(outAdMitMH)

## Use functions provided by the package coda to obtain
## quantities of interest for the density whose kernel is 'GelmanMeng'
library("coda")
draws <- as.mcmc(outAdMitMH$draws)
draws <- window(draws, start = 1001)
colnames(draws) <- c("X1", "X2")
summary(draws)
summary(draws)$stat[,3]^2 / summary(draws)$stat[,4]^2 ## RNE
```

```
plot(draws)
```

`Mit` *Mixture of Student-t Distributions*

Description

Density function or random generation for an adaptive mixture of Student-t distributions

Usage

```
dMit(theta, mit = list(), log = TRUE)
rMit(N = 1, mit = list())
```

Arguments

`theta`	matrix (of size $N \times d$, where $N, d \geq 1$) of real values.
`mit`	list containing information on the mixture approximation (see *Details*).
`log`	logical; if `log = TRUE`, returns (natural) logarithm values of the density. Default: `log = TRUE`.
`N`	number of draws (positive integer number).

Details

`dMit` returns the density values while `rMit` generates draws from a mixture of Student-t distributions.

The argument `mit` is a list containing information on the adaptive mixture of Student-t distributions. The following components must be provided:

`p` vector (of length H) of mixture probabilities.

`mu` matrix (of size $H \times d$) containing the vectors of modes (in row) of the mixture components.

`Sigma` matrix (of size $H \times d^2$) containing the scale matrices (in row) of the mixture components.

`df` degrees of freedom parameter of the Student-t components (integer number not smaller than one).

where $H(\geq 1)$ is the number of components and $d(\geq 1)$ is the dimension of the mixture approximation. Typically, `mit` is estimated by the function `AdMit`. If the `mit = list()`, a Student-t distribution located at `rep(0,d)` with scale matrix `diag(d)` and one degree of freedom parameter is used.

Value

Vector (of length N of density values, or matrix (of size `N`xd) of random draws, where $d(\geq 1)$ is the dimension of the mixture approximation.

Note

Further details and examples of the R package AdMit can be found in Ardia, Hoogerheide, van Dijk (2009a,b). See also the package vignette by typing `vignette("AdMit")`.

Please cite the package in publications. Use `citation("AdMit")`.

Author(s)

David Ardia

References

Ardia, D., Hoogerheide, L.F., van Dijk, H.K. (2009a). AdMit: Adaptive Mixture of Student-t Distributions. *The R Journal* **1**(1), pp.25-30. https://journal.R-project.org/archive/2009-1/

Ardia, D., Hoogerheide, L.F., van Dijk, H.K. (2009b). Adaptive Mixture of Student-t Distributions as a Flexible Candidate Distribution for Efficient Simulation: The R Package AdMit. *Journal of Statistical Software* **29**(3), pp.1-32. doi: 10.18637/jss.v029.i03

See Also

AdMit for fitting an adaptive mixture of Student-t distributions to a given function KERNEL, AdMitIS for importance sampling using an adaptive mixture of Student-t distributions as the importance density, AdMitMH for the independence chain Metropolis-Hastings using an adaptive mixture of Student-t distributions as the candidate density.

Examples

```
## NB : Low number of draws for speedup. Consider using more draws!
## One dimensional two components mixture of Student-t distributions
mit <- list(p = c(0.5, 0.5),
            mu = matrix(c(-2.0, 0.5), 2, 1, byrow = TRUE),
            Sigma = matrix(0.1, 2),
            df = 10)
## Generate draws from the mixture
hist(rMit(1e4, mit = mit), nclass = 100, freq = FALSE)
x <- seq(from = -5.0, to = 5.0, by = 0.01)
## Add the density to the histogram
lines(x, dMit(x, mit = mit, log = FALSE), col = "red", lwd = 2)

## Two dimensional (one component mixture) Student-t distribution
mit <- list(p = 1,
            mu = matrix(0.0, 1.0, 2.0),
            Sigma = matrix(c(1.0, 0.0, 0.0, 1.0), 1, 4),
            df = 10)
## Function used to plot the mixture in two dimensions
dMitPlot <- function(x1, x2, mit = mit)
{
  dMit(cbind(x1, x2), mit = mit, log = FALSE)
}
x1 <- x2 <- seq(from = -10.0, to = 10.0, by = 0.1)
```

```
thexlim <- theylim <- range(x1)
z <- outer(x1, x2, FUN = dMitPlot, mit = mit)
## Contour plot of the mixture
contour(x1, x2, z, nlevel = 20, las = 1,
        col = rainbow(20),
        xlim = thexlim, ylim = theylim)
par(new = TRUE)
## Generate draws from the mixture
plot(rMit(1e4, mit = mit), pch = 20, cex = 0.3,
          xlim = thexlim, ylim = theylim, col = "red", las = 1)
```

Index