Package 'relliptical'

September 15, 2021

Type Package

Title The Truncated Elliptical Family of Distributions

Version 0.1.0

Description
It generates random numbers from a truncated multivariate elliptical distribution such as Normal, Student-t, Pearson VII, Slash, Logistic, and others by specifying the density generating function. It also computes first and second moment for some particular distributions.

License GPL (>= 2)

Encoding UTF-8

Imports FuzzyNumbers.Ext.2, matrixcalc, Rcpp, RcppNumerical, Rdpack, Ryacas0, stats

RdMacros Rdpack

RoxygenNote 7.1.1

LinkingTo RcppArmadillo, Rcpp, RcppEigen, RcppNumerical

Suggests ggExtra, ggplot2, gridExtra, MomTrunc, TTmoment, tmvtnorm

NeedsCompilation yes

Author Katherine A. L. Valeriano [aut, cre] (<https://orcid.org/0000-0001-6388-4753>), Larissa Avila Matos [ctb] (<https://orcid.org/0000-0002-2635-0901>), Christian Galarza Morales [ctb] (<https://orcid.org/0000-0002-4818-6006>)

Maintainer Katherine A. L. Valeriano <katandreina@gmail.com>

Repository CRAN

Date/Publication 2021-09-15 09:10:01 UTC

R topics documented:

mvtelliptical *Mean and Variance for Truncated Multivariate Elliptical Distributions*

Description

This function approximates the mean vector and variance-covariance matrix for some particular truncated elliptical distributions through Monte Carlo integration for the truncated variables and using properties of the conditional expectation for the non-truncated variables. It supports the p-variate Normal (`Normal`), Student-t (`t`), Power Exponential (`PE`), Pearson VII (`PVII`), Slash (`Slash`), and Contaminated Normal (`CN`) distribution.

Usage

```
mvtelliptical(
  mean,
  Sigma = diag(length(mean)),
  lower = rep(-Inf, length(mean)),
  upper = rep(Inf, length(mean)),
  dist = "Normal",
  nu = NULL,
  n = 10000,
  burn.in = 0,
  thinning = 3
)
```

Arguments

`mean`	numeric vector of length p representing the location parameter.
`Sigma`	numeric positive definite matrix with dimension pxp representing the scale parameter.
`lower`	vector of lower truncation points of length p.
`upper`	vector of upper truncation points of length p.
`dist`	represents the truncated distribution to be used. The values are `Normal`, `t`, `PE`, `PVII`, `Slash` and `CN` for the truncated Normal, Student-t, Power Exponential, Pearson VII, Slash and Contaminated Normal distributions, respectively.
`nu`	additional parameter or vector of parameters depending on the density generating function. See Details.
`n`	number of Monte Carlo samples to be generated.
`burn.in`	number of samples to be discarded as a burn-in phase.
`thinning`	factor for reducing the autocorrelation of random points.

Details

This function also considers the univariate case. The argument nu is a parameter or vector of parameters depending on the density generating function (DGF). For the truncated Student-t, Power Exponential, and Slash distribution, nu is a positive number. For the truncated Pearson VII, nu is a vector with the first element greater than $p/2$ and the second element a positive number. For the truncated Contaminated Normal distribution, nu is a vector of length 2 assuming values between 0 and 1.

Value

It returns a list with three elements:

- `EY` the mean vector of length p.
- `EYY` the second moment matrix of dimensions pxp.
- `VarY` the variance-covariance matrix of dimensions pxp.

Note

The Normal distribution is a particular case of the Power Exponential distribution when nu = 1. The Student-t distribution with ν degrees of freedom results from the Pearson VII distribution when nu = ((ν+p)/2, ν).

In the Student-t distribution, if nu >= 300, the Normal case is considered. For Student-t distribution, the algorithm also supports degrees of freedom nu <= 2. For Pearson VII distribution, the algorithm supports values of m <= (p+2)/2 (first element of nu).

Author(s)

Katherine L. Valeriano, Christian E. Galarza and Larissa A. Matos

References

Fang KW (2018). *Symmetric multivariate and related distributions*. CRC Press.

Neal RM (2003). “Slice sampling.” *Annals of statistics*, 705–741.

Robert CP, Casella G (2010). *Introducing Monte Carlo Methods with R*, volume 18. Springer.

See Also

rtelliptical

Examples

```
# Truncated Student-t distribution
set.seed(5678)
mean = c(0.1, 0.2, 0.3)
Sigma = matrix(data = c(1,0.2,0.3,0.2,1,0.4,0.3,0.4,1),
               nrow=length(mean), ncol=length(mean), byrow=TRUE)

# Example 1: considering nu = 0.80 and one doubly truncated variable
a = c(-0.8, -Inf, -Inf)
```

```
b = c(0.5, 0.6, Inf)
MC11 = mvtelliptical(mean, Sigma, a, b, "t", 0.80)

# Example 2: considering nu = 0.80 and two doubly truncated variables
a = c(-0.8, -0.70, -Inf)
b = c(0.5, 0.6, Inf)
MC12 = mvtelliptical(mean, Sigma, a, b, "t", 0.80) # By default n=1e4

# Truncated Pearson VII distribution
set.seed(9876)
MC21 = mvtelliptical(mean, Sigma, a, b, "PVII", c(1.90,0.80), n=1e6) # More precision
c(MC12$EY); c(MC21$EY)
MC12$VarY;  MC21$VarY

# Truncated Normal distribution
set.seed(1234)
MC31 = mvtelliptical(mean, Sigma, a, b, "Normal", n=1e4)
MC32 = mvtelliptical(mean, Sigma, a, b, "Normal", n=1e6) # More precision
```

`rtelliptical` *Sampling Random Numbers from Truncated Multivariate Elliptical Distributions*

Description

This function generates random numbers from a truncated multivariate elliptical distribution with location parameter equal to `mean`, scale matrix `Sigma`, lower and upper truncation points `lower` and `upper` via Slice Sampling algorithm with Gibbs sampler steps.

Usage

```
rtelliptical(
  n = 10000,
  mean,
  Sigma = diag(length(mean)),
  lower = rep(-Inf, length(mean)),
  upper = rep(Inf, length(mean)),
  dist = "Normal",
  nu = NULL,
  expr = "exp(1)^(-t)/(1+exp(1)^(-t))^2",
  gFun = NULL,
  ginvFun = NULL,
  burn.in = 0,
  thinning = 1
)
```

Arguments

n	number of observations to generate. Must be an integer >= 1.
mean	numeric vector of length p representing the location parameter.
Sigma	numeric positive definite matrix with dimension pxp representing the scale parameter.
lower	vector of lower truncation points of length p.
upper	vector of upper truncation points of length p.
dist	represents the truncated distribution to be used. The values are `Normal`, `t`, `PE`, `PVII`, `Slash` and `CN` for the truncated Normal, Student-t, Power Exponential, Pearson VII, Slash and Contaminated Normal distributions, respectively.
nu	additional parameter or vector of parameters depending on the density generating function. See Details.
expr	a character with the density generating function. See Details.
gFun	an R function with the density generating function. See Details.
ginvFun	an R function with the inverse of the density generating function defined in `gFun`. See Details.
burn.in	number of samples to be discarded as a burn-in phase.
thinning	factor for reducing the autocorrelation of random points.

Details

The argument `nu` is a parameter or vector of parameters depending on the density generating function (DGF). For the truncated Student-t, Power Exponential, and Slash distribution, `nu` is a positive number. For the truncated Pearson VII, `nu` is a vector with the first element greater than $p/2$ and the second element a positive number. For the truncated Contaminated Normal distribution, `nu` is a vector of length 2 assuming values between 0 and 1.

This function also allows us to generate random points from other truncated elliptical distributions by specifying the DGF through arguments `expr` or `gFun` and making `dist` equal to `NULL`. The DGF must be non-negative and strictly decreasing on the interval `(0,Inf)`. The DGF is given as a character to argument `expr`. The notation used in `expr` needs to be understood by package `Ryacas0` and the environment of R. For example if the DGF is $g(t) = e^{-t}$, then `expr="exp(1)^(-t)"`. In this case, the algorithm tries to compute a closed expression for the inverse function of $g(t)$, a warning message is returned when it is not possible. Additionally, the function in `expr` should depend only on variable t, and any additional parameter must be a fixed value. Argument `gFun` can be accessed when `expr='NULL'`. It accepts the DGF as an R function. The inverse of the function defined in `gFun` can be provided as an R function through `ginvFun`. If `ginvFun='NULL'`, then the inverse of `gFun` is approximated numerically. In the examples, we show how to use `expr` and `gFun` to draw samples from the bivariate truncated Logistic and Kotz-type distributions. Note that the DGF of Kotz-type distribution is strictly decreasing for values of N between (2-p)/2 and 1, see Fang (2018).

Value

It returns a matrix of dimensions nxp with the random points sampled.

Note

The Normal distribution is a particular case of the Power Exponential distribution when nu = 1. The Student-t distribution with ν degrees of freedom results from the Pearson VII distribution when nu = ((ν+p)/2, ν).

Author(s)

Katherine L. Valeriano, Christian E. Galarza and Larissa A. Matos

References

Fang KW (2018). *Symmetric multivariate and related distributions*. CRC Press.

Ho HJ, Lin T, Chen H, Wang W (2012). "Some results on the truncated multivariate t distribution." *Journal of Statistical Planning and Inference*, **142**(1), 25–40.

Neal RM (2003). "Slice sampling." *Annals of statistics*, 705–741.

See Also

mvtelliptical

Examples

```
library(ggplot2)
library(ggExtra)
library(gridExtra)

# Example 1: Sampling from the Truncated Normal distribution
set.seed(1234)
mean  = c(0, 1)
Sigma = matrix(c(1,0.70,0.70,3), 2, 2)
lower = c(-2, -3)
upper = c(3, 3)
sample1 = rtelliptical(5e4, mean, Sigma, lower, upper, dist="Normal")

# Histogram and density for variable 1
ggplot(data.frame(sample1), aes(x=X1)) +
   geom_histogram(aes(y=..density..), colour="black", fill="grey", bins=15) +
   geom_density(color="red") + labs(x=bquote(X[1]), y="Density")

# Histogram and density for variable 2
ggplot(data.frame(sample1), aes(x=X2)) +
   geom_histogram(aes(y=..density..), colour="black", fill="grey", bins=15) +
   geom_density(color="red") + labs(x=bquote(X[2]), y="Density")


# Example 2: Sampling from the Truncated Logistic distribution

# Function for plotting the sample autocorrelation using ggplot2
acf.plot = function(samples){
 p = ncol(samples); n = nrow(samples); q1 = qnorm(0.975)/sqrt(n); acf1 = list(p)
 for (i in 1:p){
```

```
    bacfdf = with(acf(samples[,i], plot=FALSE), data.frame(lag, acf))
    acf1[[i]] = ggplot(data=bacfdf, aes(x=lag,y=acf)) + geom_hline(aes(yintercept=0)) +
    geom_segment(aes(xend=lag, yend=0)) + labs(x="Lag", y="ACF", subtitle=bquote(X[.(i)])) +
      geom_hline(yintercept=c(q1,-q1), color="red", linetype="twodash")
  }
  return (acf1)
}

set.seed(5678)
mean  = c(0, 0)
Sigma = matrix(c(1,0.70,0.70,1), 2, 2)
lower = c(-2, -2)
upper = c(3, 2)
# Sample autocorrelation with no thinning
sample2 = rtelliptical(1000, mean, Sigma, lower, upper, dist=NULL,
                       expr="exp(1)^(-t)/(1+exp(1)^(-t))^2")
grid.arrange(grobs=acf.plot(sample2), top="Logistic distribution with no thinning", nrow=1)

# Sample autocorrelation with thinning = 3
sample3 = rtelliptical(1000, mean, Sigma, lower, upper, dist=NULL,
                       expr="exp(1)^(-t)/(1+exp(1)^(-t))^2", thinning=3)
grid.arrange(grobs=acf.plot(sample3), top="Logistic distribution with thinning = 3", nrow=1)


# Example 3: Sampling from the Truncated Kotz-type distribution
sample4 = rtelliptical(1500, mean, Sigma, lower, upper, dist=NULL, expr=NULL,
                       gFun=function(t){ t^(-1/2)*exp(-2*t^(1/4)) })
f1 = ggplot(data.frame(sample4), aes(x=X1,y=X2)) + geom_point(size=0.50) +
     labs(x=expression(X[1]), y=expression(X[2]), subtitle="Kotz(2,1/4,1/2)")
ggMarginal(f1, type="histogram", fill="grey")
```

Index