Package ‘tmvmixnorm’

October 14, 2022

Type Package

Title Sampling from Truncated Multivariate Normal and t Distributions

Version 1.1.1

Maintainer Ting Fung (Ralph) Ma <tingfung.ma@wisc.edu>

Description Efficient sampling of truncated multivariate (scale) mixtures of normals under linear inequality constraints is nontrivial due to the analytically intractable normalizing constant. Meanwhile, traditional methods may subject to numerical issues, especially when the dimension is high and dependence is strong. Algorithms proposed by Li and Ghosh (2015) <doi:10.1080/15598608.2014.996690> are adopted for overcoming difficulties in simulating truncated distributions. Efficient rejection sampling for simulating truncated univariate normal distribution is included in the package, which shows superiority in terms of acceptance rate and numerical stability compared to existing methods and R packages. An efficient function for sampling from truncated multivariate normal distribution subject to convex polytope restriction regions based on Gibbs sampler for conditional truncated univariate distribution is provided. By extending the sampling method, a function for sampling truncated multivariate Student's t distribution is also developed. Moreover, the proposed method and computation remain valid for high dimensional and strong dependence scenarios. Empirical results in Li and Ghosh (2015) <doi:10.1080/15598608.2014.996690> illustrated the superior performance in terms of various criteria (e.g. mixing and integrated autocorrelation time).

License GPL-2

Encoding UTF-8

LazyData true

Depends R (>= 3.2.0)

Imports stats, MASS

RoxygenNote 7.0.2

NeedsCompilation no

Author Ting Fung (Ralph) Ma [cre, aut],
Sujit K. Ghosh [aut],
Yifang Li [aut]

Repository CRAN

Date/Publication 2020-09-18 18:00:02 UTC

R topics documented:

`dtuvn` *Density function of truncated univariate normal distribution*

Description

`dtuvn` calculates the probability density function (pdf) of truncated univariate normal distribution.

Usage

```
dtuvn(x, mean, sd, lower, upper)
```

Arguments

`x`	value at which density is desired.
`mean`	mean of the underlying univariate normal distribution.
`sd`	standard deviation of the underlying univariate normal distribution.
`lower`	lower bound for truncation.
`upper`	upper bound for truncation.

Value

`dtuvn` returns the density (with same dimension and type as `x`) of truncated univariate normal distribution.

Examples

```
dtuvn(x= -3:3, mean=0, sd=1 ,lower= -2, upper=2)
```

exp_acc_opt — *Acceptance rate of translated-exponential rejection sampling*

Description

`exp_acc_opt` calculates the acceptance rate of translated-exponential rejection sampling for the truncation interval (a,b).

Usage

```
exp_acc_opt(a, b)
```

Arguments

`a`	lower bound for truncation.
`b`	upper bound for truncation.

Examples

```
set.seed(1203)
exp_acc_opt(1,2)
```

exp_rej — *Translated-exponential rejection sampling*

Description

`exp_rej` is used for translated-exponential rejection sampling.

Usage

```
exp_rej(a, b = Inf, lam = "default")
```

Arguments

`a`	lower bound
`b`	upper bound
`lam`	lambda for translated-exponential only

Value

`exp_rej` returns a list `x`: sampled value; and `acc`: total number of draw used.

Examples

```
set.seed(1)
exp_rej(a=1, b=Inf)
```

---

halfnorm_acc *Acceptance rate of half-normal rejection sampling*

Description

`halfnorm_acc` calculates the acceptance rate of half-normal rejection sampling for the truncation interval (a,b).

Usage

```
halfnorm_acc(a, b)
```

Arguments

a	lower bound for truncation.
b	upper bound for truncation.

Examples

```
set.seed(1203)
halfnorm_acc(1,2)
```

---

halfnorm_rej *Half-normal rejection sampling*

Description

`halfnorm_rej` is used for half-normal rejection sampling.

Usage

```
halfnorm_rej(a, b)
```

Arguments

a	lower bound
b	upper bound

Value

`halfnorm_rej` returns a list `x`: sampled value; and `acc`: total number of draw used.

Examples

```
set.seed(1)
halfnorm_rej(a=1, b=Inf)
```

---

imp   *Rejection sampling of standardized truncated univariate normal distribution*

Description

`imp` contains a general function for rejection sampling of standardized truncated univariate normal distribution in (a,b).

Usage

```
imp(a, b)
```

Arguments

`a`	lower bound for truncation.
`b`	upper bound for truncation (must be > `a`).

Value

`imp` returns a list `x`: sampled value; and `acc`: total number of draw used.

Examples

```
imp(1,Inf) # Case 1: [a,infty)
imp(-1,1) # Case 2: 0 in [a,b], a<0<b
imp(1,2) # Case 3: [a,b], a>=0
```

imp_acc — *Acceptance rate of truncated univariate normal distribution rejection sampling*

Description

imp_acc calculates the acceptance rate of truncated univariate standardized normal distribution rejection sampling for the truncation interval (a,b).

Usage

```
imp_acc(a, b)
```

Arguments

a	lower bound for truncation.
b	upper bound for truncation.

Examples

```
imp_acc(1,Inf) # Case 1: [a,infty)
imp_acc(-1,1) # Case 2: 0 in [a,b], a<0<b
imp_acc(1,2) # Case 3: [a,b], a>=0
```

norm_acc — *Acceptance rate of normal rejection sampling*

Description

norm_acc calculates the acceptance rate of normal rejection sampling for the truncation interval (a,b).

Usage

```
norm_acc(a, b)
```

Arguments

a	lower bound for truncation.
b	upper bound for truncation.

Examples

```
set.seed(1203)
norm_acc(1,2)
```

norm_rej — *Normal rejection sampling*

Description

`norm_rej` is used for normal rejection sampling.

Usage

```
norm_rej(a, b = Inf)
```

Arguments

`a`	lower bound
`b`	upper bound

Value

`norm_rej` returns a list `x`: sampled value; and `acc`: total number of draw used.

Examples

```
set.seed(1)
norm_rej(a=1, b=Inf)
```

ptuvn — *Distribution function of truncated univariate normal distribution*

Description

`ptuvn` calculates the cumulative distribution function (cdf) of truncated univariate normal distribution.

Usage

```
ptuvn(x, mean, sd, lower, upper)
```

Arguments

`x`	value at which cdf is desired.
`mean`	mean of the underlying univariate normal distribution.
`sd`	standard deviation of the underlying univariate normal distribution.
`lower`	lower bound for truncation.
`upper`	upper bound for truncation.

Value

ptuvn returns the cumulative distribution function (with same dimension and type as x) of truncated univariate normal distribution.

Examples

```
ptuvn(x= -3:3, mean=0, sd=1 ,lower= -2, upper=2)
```

---

rtmvn *Random number generation for truncated multivariate normal distribution subject to linear inequality constraints*

Description

rtmvn simulates truncated multivariate (p-dimensional) normal distribution subject to linear inequality constraints. The constraints should be written as a matrix (D) with lower and upper as the lower and upper bounds for those constraints respectively. Note that D can be non-full rank, which generalize many traditional methods.

Usage

```
rtmvn(
  n,
  Mean,
  Sigma,
  D = diag(1, length(Mean)),
  lower,
  upper,
  int = NULL,
  burn = 10,
  thin = 1
)
```

Arguments

n	number of random samples desired (sample size).
Mean	mean vector of the underlying multivariate normal distribution.
Sigma	positive definite covariance matrix of the underlying multivariate normal distribution.
D	matrix or vector of coefficients of linear inequality constraints.
lower	vector of lower bounds for truncation.
upper	vector of upper bounds for truncation.
int	initial value vector for Gibbs sampler (satisfying truncation), if NULL then determine automatically.
burn	burn-in iterations discarded (default as 10).
thin	thinning lag (default as 1).

Value

rtmvn returns a (n*p) matrix (or vector when n=1) containing random numbers which approximately follows truncated multivariate normal distribution.

Examples

```
# Example for full rank with strong dependence
d <- 3
rho <- 0.9
Sigma <- matrix(0, nrow=d, ncol=d)
Sigma <- rho^abs(row(Sigma) - col(Sigma))

D1 <- diag(1,d) # Full rank

set.seed(1203)
ans.1 <- rtmvn(n=1000, Mean=1:d, Sigma, D=D1, lower=rep(-1,d), upper=rep(1,d),
int=rep(0,d), burn=50)

apply(ans.1, 2, summary)

# Example for non-full rank
d <- 3
rho <- 0.5
Sigma <- matrix(0, nrow=d, ncol=d)
Sigma <- rho^abs(row(Sigma) - col(Sigma))

D2 <- matrix(c(1,1,1,0,1,0,1,0,1),ncol=d)
qr(D2)$rank # 2

set.seed(1228)
ans.2 <- rtmvn(n=100, Mean=1:d, Sigma, D=D2, lower=rep(-1,d), upper=rep(1,d), burn=10)

apply(ans.2, 2, summary)
```

---

rtmvt — *Random number generation for truncated multivariate Student's t distribution subject to linear inequality constraints*

Description

rtmvt simulates truncated multivariate (p-dimensional) Student's t distribution subject to linear inequality constraints. The constraints should be written as a matrix (D) with lower and upper as the lower and upper bounds for those constraints respectively. Note that D can be non-full rank, which generalizes many traditional methods.

Usage

```
rtmvt(n, Mean, Sigma, nu, D, lower, upper, int = NULL, burn = 10, thin = 1)
```

Arguments

n	number of random samples desired (sample size).
Mean	location vector of the multivariate Student's t distribution.
Sigma	positive definite dispersion matrix of the multivariate t distribution.
nu	degrees of freedom for Student-t distribution.
D	matrix or vector of coefficients of linear inequality constraints.
lower	lower bound vector for truncation.
upper	upper bound vector for truncation.
int	initial value vector for Gibbs sampler (satisfying truncation), if NULL then determine automatically.
burn	burn-in iterations discarded (default as 10).
thin	thinning lag (default as 1).

Value

rtmvt returns a (n*p) matrix (or vector when n=1) containing random numbers which follows truncated multivariate Student-t distribution.

Examples

```
# Example for full rank
d <- 3
rho <- 0.5
nu <- 10
Sigma <- matrix(0, nrow=d, ncol=d)
Sigma <- rho^abs(row(Sigma) - col(Sigma))

D1 <- diag(1,d) # Full rank

set.seed(1203)
ans.t <- rtmvt(n=1000, Mean=1:d, Sigma, nu=nu, D=D1, lower=rep(-1,d), upper=rep(1,d),
burn=50, thin=0)

apply(ans.t, 2, summary)
```

---

rtuvn — *Random number generation for truncated univariate normal distribution*

Description

rtuvn simulates truncated univariate normal distribution within the interval.

Usage

```
rtuvn(n = 1, mean = 0, sd = 1, lower, upper)
```

Arguments

n	number of random samples desired (sample size).
mean	mean of the underlying univariate normal distribution.
sd	standard deviation of the underlying univariate normal distribution.
lower	lower bound for truncation.
upper	upper bound for truncation.

Value

`rtuvn` returns a vector of random number follows truncated univariate normal distribution.

Examples

```
set.seed(1203)
ans <- rtuvn(n=1000, mean=1, sd=2, lower=-2, upper=3)
summary(ans)

# Check if the sample matches with CDF by KS test
ks.test(ans,"ptuvn",1,2,-2,3)
```

---

unif_acc — *Acceptance rate of uniform rejection sampling*

Description

`unif_acc` calculates the acceptance rate of uniform rejection sampling for the truncation interval (a,b).

Usage

```
unif_acc(a, b)
```

Arguments

a	lower bound for truncation.
b	upper bound for truncation.

Examples

```
set.seed(1203)
unif_acc(1,2)
```

unif_rej *Uniform rejection sampling*

Description

`unif_rej` is used for uniform rejection sampling.

Usage

```
unif_rej(a, b)
```

Arguments

`a`	lower bound
`b`	upper bound

Value

`unif_rej` returns a list `x`: sampled value; and `acc`: total number of draw used.

Examples

```
set.seed(1)
unif_rej(a=1, b=2)
```

Index