

Package ‘nntmvn’

May 9, 2026

Title Draw Samples of Truncated Multivariate Normal Distributions

Version 1.3.0

Date 2025-12-03

Author Jian Cao [aut, cre],
Matthias Katzfuss [aut]

Maintainer Jian Cao <jcao2416@gmail.com>

License GPL (>= 2)

Description Draw samples from truncated multivariate normal distribution using the sequential nearest neighbor (SNN) method introduced in ``Scalable Sampling of Truncated Multivariate Normals Using Sequential Nearest-Neighbor Approximation" <doi:10.48550/arXiv.2406.17307>.

Encoding UTF-8

Imports GpGp, TruncatedNormal, RANN, dplyr, ggplot2, tidyr, R.utils,
lhs

LinkingTo Rcpp

RoxygenNote 7.3.2

NeedsCompilation yes

Repository CRAN

Date/Publication 2025-12-03 17:20:13 UTC

Contents

corr_nn	2
nntmvn	2
plot.nntmvn_1stmmt_error	3
plot.nntmvn_1stmmt_pred	3
ptmvn_check_converge	4
rptmvn	5
rtmvn	7
tmvn_check_converge	8

Index	10
--------------	-----------

corr_nn	<i>Find ordered nearest neighbors based on correlation, assuming the absolute value of the correlation is monotonically decreasing with distance. Returns an $n \times (m + 1)$ matrix, each row indicating the $m + 1$ nearest neighbors including itself.</i>
---------	---

Description

Find ordered nearest neighbors based on correlation, assuming the absolute value of the correlation is monotonically decreasing with distance. Returns an $n \times (m + 1)$ matrix, each row indicating the $m + 1$ nearest neighbors including itself.

Usage

```
corr_nn(covmat, m)
```

Arguments

covmat	the covariance matrix
m	the number of nearest neighbors

Value

an $n \times (m + 1)$ matrix

Examples

```
library(RANN)
library(nntmvn)
set.seed(123)
d <- 3
n <- 100
locs <- matrix(runif(d * n), n, d)
covparms <- c(2, 0.01, 0)
covmat <- GpGp::matern15_isotropic(covparms, locs)
m <- 10
NNarray_test <- RANN::nn2(locs, k = m + 1)[[1]]
NNarray <- nntmvn::corr_nn(covmat, m)
cat("Number of mismatch is", sum(NNarray != NNarray_test, na.rm = TRUE))
```

nntmvn

nntmvn

Description

Draw Samples of Truncated Multivariate Normal Distributions

`plot.nntmvn_1stmmt_error`*Plot function for the nntmvn_1stmmt_error class*

Description

Plot function for the nntmvn_1stmmt_error class

Usage

```
## S3 method for class 'nntmvn_1stmmt_error'  
plot(x, ...)
```

Arguments

`x` an object of the class nntmvn_1stmmt_error, returned by ptmvn_check_converge
`...` unused arguments to align with the generic interface of plot

Value

a ggplot object of class "gg" and "ggplot"

`plot.nntmvn_1stmmt_pred`*Plot function for the nntmvn_1stmmt_pred class*

Description

Plot function for the nntmvn_1stmmt_pred class

Usage

```
## S3 method for class 'nntmvn_1stmmt_pred'  
plot(x, ...)
```

Arguments

`x` an object of the class nntmvn_1stmmt_pred, returned by tmvn_check_converge
or ptmvn_check_converge
`...` unused arguments to align with the generic interface of plot

Value

a ggplot object of class "gg" and "ggplot"

ptmvm_check_converge *Check the convergence of 1st moment with m at selected indices of a PTMVN distribution with zero mean*

Description

Check the convergence of 1st moment with m at selected indices of a PTMVN distribution with zero mean

Usage

```
ptmvm_check_converge(
  y,
  cens_lb,
  cens_ub,
  covmat = NULL,
  locs = NULL,
  cov_name = NULL,
  cov_parm = NULL,
  m_vec = seq(from = 10, to = 100, by = 10),
  N = 1000,
  ind_test = NULL
)
```

Arguments

y	responses before censoring, of length n
cens_lb	cens_lb and cens_ub define the censoring region, of length n
cens_ub	cens_lb and cens_ub define the censoring region, of length n
covmat	n-by-n dense covariance matrix, either covmat or locs, cov_name, and cov_parm need to be provided
locs	location matrix n X d
cov_name	covariance function name from the GpGp package
cov_parm	parameters for the covariance function from the GpGp package
m_vec	a vector of m values (int) to be tested
N	the number of samples to generate for each test index and each m to evaluate 1st-order moment
ind_test	a vector of indices indexing the locs where we check the 1st-order moment convergence, by default, 10 random indices are used. If some test loc is not censored, the function treats it as unobserved

Value

a matrix summarizing the 1st moments evaluated at increasing m at the selected indices

Examples

```

library(GpGp)
library(nntmvn)
library(lhs)
library(ggplot2)
set.seed(123)
n <- 500
locs <- lhs::randomLHS(n, 2)
lb <- rep(-Inf, n)
ub <- rep(0, n)
covmat <- GpGp::matern15_isotropic(c(1, 0.1, 0.01), locs)
y <- as.vector(t(chol(covmat)) %*% rnorm(n))
check_obj <- ptmvn_check_converge(y, lb, ub, covmat,
                                  m_vec = seq(from = 10, to = 50, by = 10)
)
first_mmt <- check_obj$pred
plot(first_mmt)
pred_err <- check_obj$error
plot(pred_err)

```

rptmvn

Draw one sample of the underlying GP responses for a partially censored Gaussian process using sequential nearest neighbor (SNN) method

Description

Draw one sample of the underlying GP responses for a partially censored Gaussian process using sequential nearest neighbor (SNN) method

Usage

```

rptmvn(
  y,
  cens_lb,
  cens_ub,
  mask_cens,
  m = 30,
  covmat = NULL,
  locs = NULL,
  cov_name = NULL,
  cov_parm = NULL,
  NN = NULL,
  ordering = 0,
  seed = NULL
)

```

Arguments

<code>y</code>	uncensored responses of length <code>n</code> , where <code>n</code> is the number of all responses
<code>cens_lb</code>	lower bound vector for TMVN of length <code>n</code>
<code>cens_ub</code>	upper bound vector for TMVN of length <code>n</code>
<code>mask_cens</code>	mask for censored responses (also locations) of length <code>n</code>
<code>m</code>	positive integer for the number of nearest neighbors used
<code>covmat</code>	<code>n</code> -by- <code>n</code> dense covariance matrix, either <code>covmat</code> or <code>locs</code> , <code>cov_name</code> , and <code>cov_parm</code> need to be provided
<code>locs</code>	location matrix <code>n X d</code>
<code>cov_name</code>	covariance function name from the GpGp package
<code>cov_parm</code>	parameters for the covariance function from the GpGp package
<code>NN</code>	<code>n X m</code> matrix for nearest neighbors. <code>i</code> -th row is the nearest neighbor indices of <code>y_i</code> . <code>NN[i, 1]</code> should be <code>i</code>
<code>ordering</code>	0 for do not reorder, 1 for variance descending order, 2 for maximin ordering
<code>seed</code>	set seed for reproducibility

Value

a vector of length `n` representing the underlying GP responses

Examples

```
library(GpGp)
library(RANN)
library(nntmvn)
set.seed(123)
x <- matrix(seq(from = 0, to = 1, length.out = 51), ncol = 1)
cov_name <- "matern15_isotropic"
cov_parm <- c(1.0, 0.1, 0.001) #' variance, range, nugget
cov_func <- getFromNamespace(cov_name, "GpGp")
covmat <- cov_func(cov_parm, x)
y <- t(chol(covmat)) %%% rnorm(length(x))
mask <- y < 0.3
y_cens <- y
y_cens[mask] <- NA
lb <- rep(-Inf, 100)
ub <- rep(0.3, 100)
m <- 10
y_samp_mtd1 <- rptmvn(y_cens, lb, ub, mask,
  m = m, locs = x,
  cov_name = cov_name, cov_parm = cov_parm, seed = 123
)
y_samp_mtd2 <- rptmvn(y_cens, lb, ub, mask,
  m = m, covmat = covmat,
  seed = 123
)
plot(x, y_cens, ylim = range(y))
```

```

points(x[mask, ], y[mask], col = "blue")
plot(x, y_cens, ylim = range(y))
points(x[mask, ], y_samp_mtd1[mask], col = "red")
plot(x, y_cens, ylim = range(y))
points(x[mask, ], y_samp_mtd2[mask], col = "brown")

```

rtmvn	<i>Draw one sample from a truncated multivariate normal (TMVN) distribution using sequential nearest neighbor (SNN) method</i>
-------	--

Description

Draw one sample from a truncated multivariate normal (TMVN) distribution using sequential nearest neighbor (SNN) method

Usage

```

rtmvn(
  cens_lb,
  cens_ub,
  m = 30,
  covmat = NULL,
  locs = NULL,
  cov_name = NULL,
  cov_parm = NULL,
  NN = NULL,
  ordering = 0,
  seed = NULL
)

```

Arguments

cens_lb	lower bound vector for TMVN of length n
cens_ub	upper bound vector for TMVN of length n
m	positive integer for the number of nearest neighbors used
covmat	n-by-n dense covariance matrix, either covmat or locs, cov_name, and cov_parms need to be provided
locs	location matrix n X d
cov_name	covariance function name from the GpGp package
cov_parm	parameters for the covariance function from the GpGp package
NN	n X m matrix for nearest neighbors. i-th row is the nearest neighbor indices of y_i. NN[i, 1] should be i
ordering	0 for do not reorder, 1 for variance descending order, 2 for maximin ordering
seed	set seed for reproducibility

Value

a vector of length n representing the underlying GP responses

Examples

```
library(nntmvn)
library(TruncatedNormal)
set.seed(123)
x <- matrix(seq(from = 0, to = 1, length.out = 51), ncol = 1)
cov_name <- "matern15_isotropic"
cov_parm <- c(1.0, 0.1, 0.001) #' variance, range, nugget
cov_func <- getFromNamespace(cov_name, "GpGp")
covmat <- cov_func(cov_parm, x)
lb <- rep(-Inf, nrow(x))
ub <- rep(-1, nrow(x))
m <- 30
samp_SNN <- matrix(NA, 3, nrow(x))
for (i in 1:3) {
  samp_SNN[i, ] <- nntmvn::rtmvn(lb, ub, m = m, covmat = covmat, locs = x, ordering = 0)
}
samp_TN <- TruncatedNormal::rtmvnorm(3, rep(0, nrow(x)), covmat, lb, ub)
qqplot(samp_SNN, samp_TN, xlim = range(samp_SNN, samp_TN), ylim = range(samp_SNN, samp_TN))
abline(a = 0, b = 1, lty = "dashed", col = "red")
```

tmvn_check_converge *Check the convergence of 1st moment with m at selected indices of a TMVN distribution*

Description

Check the convergence of 1st moment with m at selected indices of a TMVN distribution

Usage

```
tmvn_check_converge(
  cens_lb,
  cens_ub,
  covmat = NULL,
  locs = NULL,
  cov_name = NULL,
  cov_parm = NULL,
  m_vec = seq(from = 10, to = 100, by = 10),
  N = 1000,
  ind_test = NULL
)
```

Arguments

cens_lb	lower bound vector for TMVN of length n
cens_ub	upper bound vector for TMVN of length n
covmat	n-by-n dense covariance matrix, either covmat or locs, cov_name, and cov_parm need to be provided
locs	location matrix n X d
cov_name	covariance function name from the GpGp package
cov_parm	parameters for the covariance function from the GpGp package
m_vec	a vector of m values (int) to be tested
N	the number of samples to generate for each test index and each m to evaluate 1st-order moment
ind_test	a vector of indices indexing the locs where we check the 1st-order moment convergence, by default, 10 random indices are used

Value

a matrix summarizing the 1st moments evaluated at increasing m at the selected indices

Examples

```

library(GpGp)
library(nntmvn)
library(lhs)
library(ggplot2)
set.seed(123)
n <- 500
locs <- lhs::randomLHS(n, 2)
lb <- rep(-Inf, n)
ub <- rep(0, n)

# using covariance matrix
covmat <- GpGp::matern15_isotropic(c(1, 0.1, 0.001), locs)
first_mmt <- tmvn_check_converge(lb, ub, covmat,
  m_vec = seq(from = 10, to = 50, by = 10)
)
plot(first_mmt)

# using locs, cov_name, and cov_parm
cov_name <- "matern15_isotropic"
cov_parm <- c(1, 0.1, 0.001)
first_mmt <- tmvn_check_converge(lb, ub,
  locs = locs, cov_name = cov_name, cov_parm = cov_parm,
  m_vec = seq(from = 10, to = 50, by = 10)
)
plot(first_mmt) + theme(text = element_text(size = 14))

```

Index

`corr_nn`, 2

`nntmvn`, 2

`plot.nntmvn_1stmmt_error`, 3

`plot.nntmvn_1stmmt_pred`, 3

`ptmvn_check_converge`, 4

`rptmvn`, 5

`rtmvn`, 7

`tmvn_check_converge`, 8