

Package ‘nntmvn’

July 27, 2024

Title Draw Samples of Truncated Multivariate Normal Distributions

Version 1.0.0

Date 2024-07-15

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

Maintainer Jian Cao <jcao2416@gmail.com>

License GPL (>= 2)

Description Use the sequential nearest neighbor (SNN) method introduced in Jian Cao and Matthias Katzfuss (2024) <[doi:10.48550/arXiv.2406.17307](https://doi.org/10.48550/arXiv.2406.17307)> to draw samples from the truncated multivariate normal (TMVN) distributions.

Encoding UTF-8

Imports GpGp, TruncatedNormal

Suggests RANN

RoxygenNote 7.3.2

NeedsCompilation no

Repository CRAN

Date/Publication 2024-07-27 16:40:02 UTC

Contents

rtmvn_snn	2
Index	4

rtmvn_snn	<i>Simulate the underlying GP responses for censored responses using nearest neighbors</i>
-----------	--

Description

Simulate the underlying GP responses for censored responses using nearest neighbors

Usage

```
rtmvn_snn(
  y,
  cens_lb,
  cens_ub,
  mask_cens,
  NN,
  locs,
  cov_name,
  cov_parm,
  covmat = NULL,
  seed = NULL
)
```

Arguments

y	uncensored responses of length n, where n is the number of all responses
cens_lb	lower bound vector for TMVN of length n
cens_ub	upper bound vector for TMVN of length n
mask_cens	mask for censored responses (also locations) of length n
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
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
covmat	(optional) n-by-n dense covariance matrix, not needed if locs, cov_name, and cov_parms are provided
seed	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
NN <- RANN::nn2(x, k = m + 1)[[1]]
y_samp <- rtmvn_snn(y_cens, lb, ub, mask, NN, x, cov_name, cov_parm)

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[mask], col = "red")
```

Index

rtmvt_snn, 2