

# Package ‘tlrmvnmvt’

April 6, 2020

**Type** Package

**Title** Low-Rank Methods for MVN and MVT Probabilities

**Version** 1.1.0

**Date** 2020-03-26

**Author** Jian Cao, Marc Genton, David Keyes, George Turkiyyah

**Maintainer** Jian Cao <jian.cao@kaust.edu.sa>

**Description** Implementation of the classic Genz algorithm and a novel tile-low-rank algorithm for computing relatively high-dimensional multivariate normal (MVN) and Student-t (MVT) probabilities.

References used for this package:

Foley, James, Andries van Dam, Steven Feiner, and John Hughes.

“Computer Graphics: Principle and Practice”. Addison-Wesley

Publishing Company. Reading, Massachusetts (1987, ISBN:0-201-84840-6 1);

Genz, A., “Numerical computation of multivariate

normal probabilities,” Journal of Computational and

Graphical Statistics, 1, 141-149 (1992) <doi:10.1080/10618600.1992.10477010>;

Cao, J., Genton, M. G., Keyes, D. E., & Turkiyyah, G. M. “Exploiting Low

Rank Covariance Structures for Computing High-Dimensional Normal and Student-

t Probabilities” (2019) <https://marcgenton.github.io/2019.CGKT.manuscript.pdf>.

**License** GPL-2

**Depends** R (>= 3.6.0)

**Suggests** mvtnorm (>= 1.0.11)

**Imports** Rcpp (>= 1.0.1)

**LinkingTo** Rcpp (>= 1.0.1), RcppEigen (>= 0.3.3.5.0), BH (>= 1.69.0.1)

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2020-04-06 10:00:03 UTC

## R topics documented:

tlrmvnmvt-package . . . . . 2

algorithm . . . . .	3
pmvn . . . . .	3
pmvt . . . . .	5
zorder . . . . .	7
<b>Index</b>	<b>9</b>

---

tlrmvnmvt-package	<i>Low-Rank Methods for MVN and MVT Probabilities</i>
-------------------	---

---

## Description

Implementation of the classic Genz algorithm and a novel tile-low-rank algorithm for computing relatively high-dimensional multivariate normal (MVN) and Student-t (MVT) probabilities. References used for this package: Foley, James, Andries van Dam, Steven Feiner, and John Hughes. "Computer Graphics: Principle and Practice". Addison-Wesley Publishing Company. Reading, Massachusetts (1987, ISBN:0-201-84840-6 1); Genz, A., "Numerical computation of multivariate normal probabilities," *Journal of Computational and Graphical Statistics*, 1, 141-149 (1992) <doi:10.1080/10618600.1992.10477010>; Cao, J., Genton, M. G., Keyes, D. E., & Turkiyyah, G. M. "Exploiting Low Rank Covariance Structures for Computing High-Dimensional Normal and Student- t Probabilities" (2019) <<https://marcgenton.github.io/2019.CGKT.manuscript.pdf>>.

## Details

Implementation of the classic Genz algorithm and a novel tile-low-rank algorithm for computing relatively high-dimensional multivariate normal and Student-t probabilities. For the Genz's algorithm (GenzBretz), we apply a univariate reordering preconditioner and for the tile-low-rank algorithms (TLRQMC), we apply a recursive block reordering preconditioner. The GenzBretz methods are different from their counterparts in the 'mvtnorm' package in that the 'tlrmvnmvt' package can accept any problem dimension and return the result in the log2 fashion, which is useful when the true probability is smaller than the machine precision. The TLRQMC algorithms can compute the probabilities up to tens of thousands of dimensions with the low-rank representation. However, this category of algorithms requires the existence of the low-rank property in the off-diagonal blocks of size  $m$ . The zorder function implements Morton's order in the 2D plane, which enhances the low-rank property of the produced covariance matrices.

Package functions: pmvn, pmvt, zorder

## Author(s)

Jian Cao, Marc Genton, David Keyes, George Turkiyyah

Maintainer: Jian Cao <[jian.cao@kaust.edu.sa](mailto:jian.cao@kaust.edu.sa)>

## References

Cao, J., Genton, M. G., Keyes, D. E., & Turkiyyah, G. M. (2019). "Exploiting Low Rank Covariance Structures for Computing High-Dimensional Normal and Student- t Probabilities"

---

 algorithm

*Parameters for the Quasi-Monte Carlo sampling*


---

**Description**

The two functions return objects containing the parameters for the dense-matrix based Quasi-Monte Carlo method and the tile-low-rank-matrix based Quasi-Monte Carlo method, respectively.

**Usage**

```
GenzBretz(N = 499)
TLRQMC(N = 499, m = 64, eps1 = 1e-4)
```

**Arguments**

N	an integer, specifying the number of per-batch Monte Carlo samples. The total number of Monte Carlo samples is 20 X N
m	an integer, specifying the block size for the tile-low-rank methods
eps1	numeric value, specifying the truncation level for the tile-low-rank methods

**Value**

Return an object of the class "GenzBretz" or "TLRQMC", which is used as the parameter of the [pmvn](#) and the [pmvt](#) functions.

**Author(s)**

Jian Cao, Marc Genton, David Keyes, George Turkiyyah

---

 pmvn

*Quasi-Monte Carlo method for multivariate normal probabilities*


---

**Description**

Compute multivariate normal probabilities with the dense-matrix based Quasi-Monte Carlo method and the tile-low-rank-matrix based Quasi-Monte Carlo method.

**Usage**

```
pmvn(lower = -Inf, upper = Inf, mean = 0, sigma = NULL,
      uselog2 = FALSE, algorithm = GenzBretz(), ...)
```

**Arguments**

lower	lower integration limits, a numeric vector of length n
upper	upper integration limits, a numeric vector of length n
mean	the mean parameter, a numeric vector of length n
sigma	the covariance matrix of dimension n
uselog2	whether return the result as the logarithm to the base 2
algorithm	an object of class <code>GenzBretz</code> or <code>TLRQMC</code> defining the hyper parameters of this algorithm
...	additional parameters used to construct 'sigma' when it is not given: <ul style="list-style-type: none"> <li>• <code>geom</code> matrix of dimension n-by-2, specifying n spatial locations in the 2D unit square</li> <li>• <code>type</code> the name of the covariance kernel, a string. Currently, only the Matern covariance function, e.g., "matern", is supported. Not case-sensitive. It should be given when 'sigma' is not given</li> <li>• <code>para</code> the parameter for the covariance kernel, a numeric vector. When 'type' is "matern", the length of 'para' should be 4, representing the scale, range, smoothness, and nugget parameters of the covariance function. It should be given when 'sigma' is not given</li> </ul>

**Details**

When 'algorithm' is of the class 'GenzBretz', the Quasi-Monte Carlo sampling described in Genz, A. (1992) is used. When 'algorithm' is of the class 'TLRQMC', the Quasi-Monte Carlo sampling with the tile-low-rank representation of the covariance matrix, described in Cao et al. (2020), is used. When 'sigma', is given, 'geom', 'type', and 'para' are not used. Otherwise, a covariance matrix is created with the information from 'geom', 'type', and 'para'.

**Value**

When 'uselog2' is set FALSE, the function returns the estimated probability with one attribute of the estimation error. When 'uselog2' is set TRUE, the function only returns the estimated log-probability to the base 2. This is useful when the estimated probability is smaller than the machine precision.

**Author(s)**

Jian Cao, Marc Genton, David Keyes, George Turkiyyah

**References**

Genz, A. (1992), "Numerical computation of multivariate normal probabilities," *Journal of Computational and Graphical Statistics*, 1, 141-149. Cao, J., Genton, M. G., Keyes, D. E., & Turkiyyah, G. M. (2019). "Exploiting Low Rank Covariance Structures for Computing High-Dimensional Normal and Student- t Probabilities"

**Examples**

```

n = 225
set.seed(0)
a = rep(-10, n)
b = rnorm(n, 3, 2)
m = 15
eps1 = 1e-4
vec1 = 1 : m
vec2 = rep(1, m)
geom = cbind(kronecker(vec1, vec2), kronecker(vec2, vec1))
geom = geom / m
beta = 0.3
idx = zorder(geom)
geom = geom[idx, ]
a = a[idx]
b = b[idx]
distM = as.matrix(dist(geom))
covM = exp(-distM / beta)
pmvn(lower = a, upper = b, mean = 2, sigma = covM, uselog2 = FALSE,
      algorithm = GenzBretz(N = 521))
pmvn(lower = a, upper = b, mean = 2, uselog2 = TRUE, geom = geom,
      kernelType = "matern", para = c(1.0, 0.3, 0.5, 0.0))
pmvn(lower = a, upper = b, mean = 2, sigma = covM, uselog2 = FALSE,
      algorithm = TLRQMC(N = 521, m = m, eps1 = eps1))
pmvn(lower = a, upper = b, mean = 2, uselog2 = TRUE, geom = geom,
      algorithm = TLRQMC(N = 521, m = m, eps1 = eps1),
      kernelType = "matern", para = c(1.0, 0.3, 0.5, 0.0))

```

pmvt

*Quasi-Monte Carlo method for Student- $t$  probabilities***Description**

Compute multivariate Student- $t$  probabilities with the dense-matrix based Quasi-Monte Carlo method and the tile-low-rank-matrix based Quasi-Monte Carlo method.

**Usage**

```

pmvt(lower = -Inf, upper = Inf, delta = 0, df = 1, sigma = NULL,
      uselog2 = FALSE, algorithm = GenzBretz(),
      type = "Kshirsagar", ...)

```

**Arguments**

lower	lower integration limits, a numeric vector of length n
upper	upper integration limits, a numeric vector of length n
delta	the vector of noncentrality parameters of length n, for type = "shifted" delta specifies the mode

df	a positive numeric value denoting the degrees of freedom
sigma	the covariance matrix of dimension n
uselog2	whether return the result as the logarithm to the base 2
algorithm	an object of class <code>GenzBretz</code> or <code>TLRQMC</code> defining the hyper parameters of this algorithm
type	type of the noncentral multivariate $t$ distribution to be computed. <code>'type' = "Kshirsagar"</code> corresponds to formula (1.4) in Genz and Bretz (2009). <code>'type' = "shifted"</code> corresponds to the formula right before formula (1.4) in Genz and Bretz (2009)
...	additional parameters used to construct <code>'sigma'</code> when it is not given: <ul style="list-style-type: none"> <li>• <code>geom</code> matrix of dimension n-by-2, specifying n spatial locations in the 2D unit square</li> <li>• <code>type</code> the name of the covariance kernel, a string. Currently, only the Matern covariance function, e.g., "matern", is supported. Not case-sensitive. It should be given when <code>'sigma'</code> is not given</li> <li>• <code>para</code> the parameter for the covariance kernel, a numeric vector. When <code>'type'</code> is "matern", the length of <code>'para'</code> should be 4, representing the scale, range, smoothness, and nugget parameters of the covariance function. It should be given when <code>'sigma'</code> is not given</li> </ul>

### Details

When `'algorithm'` is of the class `'GenzBretz'`, the Quasi-Monte Carlo sampling described in Genz, A. (1992) is used. When `'algorithm'` is of the class `'TLRQMC'`, the Quasi-Monte Carlo sampling with the tile-low-rank representation of the covariance matrix, described in Cao et al. (2020), is used. When `'sigma'`, is given, `'geom'`, `'type'`, and `'para'` are not used. Otherwise, a covariance matrix is created with the information from `'geom'`, `'type'`, and `'para'`.

### Value

When `'uselog2'` is set FALSE, the function returns the estimated probability with one attribute of the estimation error. When `'uselog2'` is set TRUE, the function only returns the estimated log-probability to the base 2. This is useful when the estimated probability is smaller than the machine precision.

### Author(s)

Jian Cao, Marc Genton, David Keyes, George Turkiyyah

### References

Genz, A. (1992), "Numerical computation of multivariate normal probabilities," *Journal of Computational and Graphical Statistics*, 1, 141-149. Genz, A. and Bretz, F. (2009), *Computation of Multivariate Normal and t Probabilities*. Lecture Notes in Statistics, Vol. 195. Springer-Verlag, Heidelberg. Cao, J., Genton, M. G., Keyes, D. E., & Turkiyyah, G. M. (2019). "Exploiting Low Rank Covariance Structures for Computing High-Dimensional Normal and Student- t Probabilities"

**Examples**

```

n = 225
set.seed(0)
a = rep(-10, n)
b = rnorm(n, 3, 2)
m = 15
eps1 = 1e-4
vec1 = 1 : m
vec2 = rep(1, m)
geom = cbind(kronecker(vec1, vec2), kronecker(vec2, vec1))
geom = geom / m
beta = 0.3
idx = zorder(geom)
geom = geom[idx, ]
a = a[idx]
b = b[idx]
distM = as.matrix(dist(geom))
covM = exp(-distM / beta)
df = 10
pmvt(lower = a, upper = b, delta = 2, df = df,
      sigma = covM, uselog2 = FALSE, algorithm = GenzBretz(N = 521),
      type = "Kshirsagar")
pmvt(lower = a, upper = b, delta = 2, df = df,
      uselog2 = TRUE, type = "shifted", geom = geom,
      kernelType = "matern", para = c(1.0, 0.3, 0.5, 0.0))
pmvt(lower = a, upper = b, delta = 2, df = df,
      sigma = covM, uselog2 = FALSE,
      algorithm = TLRQMC(N = 521, m = m, eps1 = eps1), type = "Kshirsagar")
pmvt(lower = a, upper = b, delta = 2, df = df,
      uselog2 = TRUE, type = "shifted", geom = geom,
      algorithm = TLRQMC(N = 521, m = m, eps1 = eps1),
      kernelType = "matern", para = c(1.0, 0.3, 0.5, 0.0))

```

---

zorder

*Index locations on the 2D plane*


---

**Description**

Index a set of locations of the 2D plane with a Z-curve, when they are scaled into the unit square. The goal of this function is to make locations close in geometry also close in the index. When partitioned into clusters, the inter-cluster correlation is more likely to be low-rank.

**Usage**

```
zorder(geom)
```

**Arguments**

**geom** 2D geometry, each row represents the location of a variable. The geometry should be scaled into the unit square, (0,1) X (0,1)

**Value**

A vector of indices based on the Z-curve.

**Author(s)**

Jian Cao, Marc Genton, David Keyes, George Turkiyyah

**References**

Foley, James, Andries van Dam, Steven Feiner, and John Hughes. "Computer Graphics: Principle and Practice". Addison-Wesley Publishing Company. Reading, Massachusetts: 1987. pages 870-871

**Examples**

```
n = 333
set.seed(0)
geom = matrix(runif(n*2), n, 2)
idx = zorder(geom)
idx
```



# Index

algorithm, 3

GenzBretz, 4, 6

GenzBretz (algorithm), 3

pmvn, 3, 3

pmvt, 3, 5

t1rmvnmvt (t1rmvnmvt-package), 2

t1rmvnmvt-package, 2

TLRQMC, 4, 6

TLRQMC (algorithm), 3

zorder, 7