Package 'rvMF'

Title Fast Generation of von Mises-Fisher Distributed Pseudo-Random Vectors

Version 0.0.8

Description Generates pseudo-random vectors that follow an arbitrary von Mises-Fisher distribution on a sphere. This method is fast and efficient when generating a large number of pseudo-random vectors. Functions to generate random variates and compute density for the distribution of an inner product between von Mises-Fisher random vector and its mean direction are also provided.

URL https://github.com/seungwoo-stat/rvMF

BugReports https://github.com/seungwoo-stat/rvMF/issues

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

LinkingTo Rcpp

Imports Bessel (>= 0.6-0), Rcpp (>= 1.0.10), Rfast (>= 2.0.6), scModels (>= 1.0.4)

NeedsCompilation yes

Author Seungwoo Kang [aut, cre] (<https://orcid.org/0000-0001-8082-0794>), Hee-Seok Oh [aut]

Maintainer Seungwoo Kang <kangsw0401@snu.ac.kr>

Repository CRAN

Date/Publication 2023-11-07 07:20:02 UTC

R topics documented:

---

rvMF   *von Mises-Fisher Distributed Pseudo-Random Vector Generator*

Description

`rvMF()` generates von-Mises Fisher distributed pseudo-random vectors, without resorting to the rejection-based sampling method proposed by Wood (1994). This function partly uses the code from the function `Rfast::rvmf()` and the article Marsaglia et al. (2004).

Usage

```
rvMF(n, mu, k)
```

Arguments

n	number of pseudo-random vectors to generate.
mu	mean direction.
k	concentration parameter. $k \geq 0$.

Value

matrix where each row independently follows the specified von Mises-Fisher distribution. The number of columns equals the length of `mu`, and the number of rows equals `n` for `rvMF`.

References

K. V. Mardia and P. E. Jupp. *Directional Statistics*, volume 494. John Wiley & Sons, Chichester, 1999.

G. Marsaglia, W. W. Tsang, and J. Wang. Fast generation of discrete random variables. *Journal of Statistical Software*, 11(3):1–11, 2004.

M. Papadakis, M. Tsagris, M. Dimitriadis, S. Fafalios, I. Tsamardinos, M. Fasiolo, G. Borboudakis, J. Burkardt, C. Zou, K. Lakiotaki, and C. Chatzipantsiou. *Rfast: A Collection of Efficient and Extremely Fast R Functions*, 2022. https://CRAN.R-project.org/package=Rfast. R package version 2.0.6.

A. T. Wood. Simulation of the von Mises Fisher distribution. *Communications in Statistics– Simulation and Computation*, 23(1):157–164, 1994.

See Also

rvMFangle(), dvMFangle(), Rfast::rvmf().

Examples

```
rvMF(10, c(0,0,1), 10)
rvMF(10, c(1,1)/sqrt(2), 0)
```

| vMFangle | *Inner Product of von Mises-Fisher Random Vector and Mean Direction* |

Description

These functions provide information about the distribution of an inner product between von Mises-Fisher random vector and its mean direction. Specifically, if X follows a von Mises-Fisher distribution with mean direction μ, the inner product $X'\mu$ will be a random variable following some distribution. See page 170 of Mardia and Jupp (1999). `rvMFangle()` generates random variates, and `dvMFangle` gives the density from this distribution. This function partly uses the code from the article Marsaglia et al. (2004).

Usage

```
rvMFangle(n, p, kappa)

dvMFangle(r, p, kappa)
```

Arguments

n	number of random vectors to generate.
p	dimension of the sphere. i.e., S^{p-1}, $p \geq 2$.
kappa	concentration parameter. $kappa > 0$. Setting kappa=0 may cause errors.
r	vector of quantiles. $-1 \leq r \leq 1$.

Value

- `rvMFangle()` returns a vector whose components independently follow the aforementioned distribution. The length of the result is determined by n for `rvMFangle()`.

- `dvMFangle()` returns a vector of density function value. The length of the result is determined by the length of r for `dvMFangle()`.

References

K. V. Mardia and P. E. Jupp. *Directional Statistics*, volume 494. John Wiley & Sons, Chichester, 1999.

G. Marsaglia, W. W. Tsang, and J. Wang. Fast generation of discrete random variables. *Journal of Statistical Software*, 11(3):1–11, 2004.

See Also

[rvMF()](#) wrapper of `rvMFangle()`.

Examples

```
rvMFangle(10, 2, 10)
rvMFangle(10, 3, 0.1)
dvMFangle(seq(-1,1,by=0.01), 2, 10)
dvMFangle(seq(0,1,by=0.01), 3, 0.1)
```

Index