Package ‘DRAYL’

January 20, 2025

Version 1.0

Title Computation of Rayleigh Densities of Arbitrary Dimension

Author Martin Wiegand

Maintainer Martin Wiegand <Martin.Wiegand@manchester.ac.uk>

Depends R (>= 3.0.1)

Description We offer an implementation of the series representation put forth in ``A series representation for multidimensional Rayleigh distributions'' by Wiegand and Nadarajah <DOI:10.1002/dac.3510>. Furthermore we have implemented an integration approach proposed by Beaulieu et al. for 3 and 4-dimensional Rayleigh densities (Beaulieu, Zhang, ``New simplest exact forms for the 3D and 4D multivariate Rayleigh PDFs with applications to antenna array geometrics'', <DOI:10.1109/TCOMM.2017.2709307>).

License GPL-2

Imports stats,pracma,RConics,rmutil,cubature

Encoding UTF-8

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2019-08-21 08:20:07 UTC

Contents

alphamatrix — *Computation of Alpha coefficient matrix*

Description

The alpha matrix is a necessary intermediate step in the series expansion approach. It lists the different parameter combinations necessary for the series expansion.

Usage

```
alphamatrix(n)
```

Arguments

n	Distribution dimension.

Value

Returns a `n-1` dimensional matrix that contains the permutations of all indeces.

Examples

```
alphamatrix(3)
```

btcol — *Auxilliary function computing factors.*

Description

Auxilliary function, that evaluates coefficents for elements of the indices matrix.

Usage

```
btcol(col)
```

Arguments

col	Variables t,a and j to be combined

Value

Coefficients need to be computed for the entire permutation matrix of indices, this is the columnwise evaluation based on `t`,`a` and `j`.

btprod — *Auxilliary function computing intermediate products.*

Description

Auxilliary function. Based on the results of the `btcol` the row wise results are computed.

Usage

```
btprod(t,a,Jstar)
```

Arguments

`t`	Index number.
`a`	The respective Alpha matrix value.
`Jstar`	Matrix of the j-star indeces of the series expansion.

Value

Returns the row-wise multiplication of the coefficients based on the indeces `j`.

drayl3D — *Three dimensional Rayleigh density by series expansion*

Description

Returns a 3D Rayleigh density for arbitrary covariance values. The resulting function can then be evaluated at arbitrary points.

Usage

```
drayl3D(dK,Ccomp,lim)
```

Arguments

`dK`	Determinant of the covariance matrix.
`Ccomp`	"Compressed" cofactor matrix, leaving out zero value entries.
`lim`	Number of series terms.

Value

The 3D Rayleigh density for the compressed cofactor matrix `Ccomp` of the covariance matrix. The function can then be evaluated for 3-dimensional vectors `r`.

Examples

```
library("RConics")

# Matrix
K3 = matrix(0,nrow = 6,ncol = 6)
sigma3 = sqrt(c(0.5,1,1.5))
diag(K3) = c(0.5,0.5,1,1,1.5,1.5)

# rho_12 rho_13 rho_23
rho3<-c(0.9,0.8,0.7)

K3[1,3]=K3[3,1]=K3[2,4]=K3[4,2]=sigma3[1]*sigma3[2]*rho3[1]
K3[1,5]=K3[5,1]=K3[2,6]=K3[6,2]=sigma3[1]*sigma3[3]*rho3[2]
K3[3,5]=K3[5,3]=K3[4,6]=K3[6,4]=sigma3[2]*sigma3[3]*rho3[3]


C3=adjoint(K3)
n = nrow(K3)/2
Ccomp3<-C3[seq(1,(2*n-1),2),][,seq(1,(2*n-1),2)]
dK3<-det(K3)


pdf3D<-drayl3D(dK = dK3, Ccomp = Ccomp3, lim = 3)

pdf3D(rep(1,3))
```

`drayl4D` *Four dimensional Rayleigh density by series expansion*

Description

Returns a 4D Rayleigh density for arbitrary covariance values. The resulting function can then be evaluated at arbitrary points.

Usage

```
drayl4D(dK,Ccomp,lim)
```

Arguments

dK	Determinant of the covariance matrix.
Ccomp	"Compressed" cofactor matrix, leaving out zero value entries.
lim	Number of series terms.

Value

The 4D Rayleigh density for the compressed cofactor matrix `Ccomp` of the covariance matrix. The function can then be evaluated for 4-dimensional vectors `r`.

Examples

```
library("RConics")

K4 = matrix(0,nrow = 8,ncol = 8)
sigma4 = sqrt(c(0.5,1,1.5,1))
rho4<-c(0.7,0.75,0.8,0.7,0.75,0.7)

K4[1,1]=K4[2,2]=sigma4[1]^2
K4[3,3]=K4[4,4]=sigma4[2]^2
K4[5,5]=K4[6,6]=sigma4[3]^2
K4[7,7]=K4[8,8]=sigma4[4]^2

K4[1,3]=K4[3,1]=K4[2,4]=K4[4,2]=sigma4[1]*sigma4[2]*rho4[1]
K4[1,5]=K4[5,1]=K4[2,6]=K4[6,2]=sigma4[1]*sigma4[3]*rho4[2]
K4[1,7]=K4[7,1]=K4[2,8]=K4[8,2]=sigma4[1]*sigma4[4]*rho4[3]
K4[3,5]=K4[5,3]=K4[4,6]=K4[6,4]=sigma4[2]*sigma4[3]*rho4[4]
K4[3,7]=K4[7,3]=K4[4,8]=K4[8,4]=sigma4[2]*sigma4[4]*rho4[5]
K4[5,7]=K4[7,5]=K4[8,6]=K4[6,8]=sigma4[3]*sigma4[4]*rho4[6]

C4=adjoint(K4)
n = nrow(K4)/2
Ccomp4<-C4[seq(1,(2*n-1),2),][,seq(1,(2*n-1),2)]
dK4<-det(K4)

pdf4D<-drayl4D(dK = dK4, Ccomp = Ccomp4, lim = 3)
pdf4D(rep(1,4))
```

`drayl_int3D` *Three Dimensional Rayleigh Density by Integration*

Description

A three dimensional Rayleigh density by integration.

Usage

```
drayl_int3D(r,omega,sigma,cor,method)
```

Arguments

r	Evaluation point.
omega	Omega construct necessary for the Integration method.
sigma	Variances of the signals.
cor	Correlation structure.
method	Integration methods, either "Kronrod","Clenshaw","Simpson","Romberg","TOMS614" or "mixed".

Value

Evaluates the 3D Rayleigh density at the point r, for the values omega,sigma and cor as specified by Bealieu's method.

Examples

```
# Matrix
K3 = matrix(0,nrow = 6,ncol = 6)
sigma3 = sqrt(c(0.5,1,1.5))
diag(K3) = c(0.5,0.5,1,1,1.5,1.5)

# rho_12 rho_13 rho_23
rho3<-c(0.9,0.8,0.7)

K3[1,3]=K3[3,1]=K3[2,4]=K3[4,2]=sigma3[1]*sigma3[2]*rho3[1]
K3[1,5]=K3[5,1]=K3[2,6]=K3[6,2]=sigma3[1]*sigma3[3]*rho3[2]
K3[3,5]=K3[5,3]=K3[4,6]=K3[6,4]=sigma3[2]*sigma3[3]*rho3[3]

cor3 = rho3

mat<-diag(3)
mat[1,2]=mat[2,1]=cor3[1]
mat[1,3]=mat[3,1]=cor3[2]
mat[2,3]=mat[3,2]=cor3[3]

omega3=mat

drayl_int3D(c(1,1,1),omega = omega3,sigma = sigma3,cor = cor3, method = "Romberg")
```

---

drayl_int4D *Four Dimensional Rayleigh Density by Integration*

Description

A four dimensional Rayleigh density by integration.

Usage

```
drayl_int4D(r,omega,sigma,cor,method)
```

Arguments

r	Evaluation point.
omega	Omega construct necessary for the Integration method.
sigma	Variances of the signals.
cor	Correlation structure.
method	Integration methods, either "Romberg","Cubature" or "Quadrature".

Value

Evaluates the 4D Rayleigh density at the point `r`, for the values `omega`,`sigma` and `cor` as specified by Bealieu's method.

Examples

```
library("RConics")

K4 = matrix(0,nrow = 8,ncol = 8)
sigma4 = sqrt(c(0.5,1,1.5,1))
rho4<-c(0.7,0.75,0.8,0.7,0.75,0.7)

K4[1,1]=K4[2,2]=sigma4[1]^2
K4[3,3]=K4[4,4]=sigma4[2]^2
K4[5,5]=K4[6,6]=sigma4[3]^2
K4[7,7]=K4[8,8]=sigma4[4]^2

K4[1,3]=K4[3,1]=K4[2,4]=K4[4,2]=sigma4[1]*sigma4[2]*rho4[1]
K4[1,5]=K4[5,1]=K4[2,6]=K4[6,2]=sigma4[1]*sigma4[3]*rho4[2]
K4[1,7]=K4[7,1]=K4[2,8]=K4[8,2]=sigma4[1]*sigma4[4]*rho4[3]
K4[3,5]=K4[5,3]=K4[4,6]=K4[6,4]=sigma4[2]*sigma4[3]*rho4[4]
K4[3,7]=K4[7,3]=K4[4,8]=K4[8,4]=sigma4[2]*sigma4[4]*rho4[5]
K4[5,7]=K4[7,5]=K4[8,6]=K4[6,8]=sigma4[3]*sigma4[4]*rho4[6]

sigma4 = c(sqrt(c(K4[1,1],K4[3,3],K4[5,5],K4[7,7])))

cor4 = c(K4[1,3]/(sigma4[1]*sigma4[2]),
         K4[1,5]/(sigma4[1]*sigma4[3]),
         K4[1,7]/(sigma4[1]*sigma4[4]),
         K4[3,5]/(sigma4[2]*sigma4[3]),
         K4[3,7]/(sigma4[2]*sigma4[4]),
         K4[5,7]/(sigma4[3]*sigma4[4]))

omega4=omega4<-matrix(data = c(1,cor4[1],cor4[2],cor4[3],cor4[1],1,cor4[4],
                  cor4[5],cor4[2],cor4[4],1,cor4[6],cor4[3],cor4[5],cor4[6],1),nrow = 4)

drayl_int4D(c(1,1,1,1),omega = omega4,sigma = sigma4,cor = cor4, method = "Cubature")
```

`zerooneoutput` *Non-zero value determination*

Description

Determines the contribution of sum terms, based on the index `j`, `rho` and the matrix `A`.

Usage

```
zerooneoutput(j,rho,A)
```

Arguments

`j`	Vector of j indeces.
`rho`	Vector of the rho index.
`A`	Alpha matrix.

Value

Either 0 or 1, computes the integral contribution based on the alphamatrix `A`.

Examples

```
A = alphamatrix(3)
zerooneoutput(c(0,0,0),c(-1,-1,-1),A)
```

Index