**Documentation for CDFandPDFmvna\_v3.R**

1. **Main functions**

The calls to the approximation procedures have the following structure:

cdfmvnanalytic(mu,cov,x,s); analytically approximated CDF of multivariate normal

pdfmvnanalytic(mu,cov,x,s): Gradient of approximated CDF

GLOBALS:

.covarr = 1 implies cov is a covariance matrix; .covarr=0 means cov is a correlation matrix

.perms = n means n permutations of abscissae will be used in the Switzer, Solow, Joe analytic approach, n=1 means only one permutation will be used; For all non-SSJ methods, only one sequence (based on global .optimal) will be used, and .perms is irrelevant

.optimal = 0 implies that non-SSJ methods will be based on simple ascending order of abscissae

= 1 means non-SSJ methods will be based on an ordering following the GGE approach

= 2 implies that non-SSJ methods will be based on random ordering of the abscissae

= 3 implies the abscissae are used in the same order as given.

The value of .optimal does not matter for the SSJ method.

IMPORTANT NOTE: For the TVBS method, only the .optimal=0, 2, or 3 values must be used to get correct analytic gradients, because the TVBS method already is based on an approximate quadrivariate CDF function evaluation; this requirement is already taken care of in the code, by putting .optimal=0 (if .optimal is provided as 1) if the TVBS method is used.

.method = "SSJ" - Switzer, Solow, and Joe Method

"TG" - Trinh and Genz's univariate conditioning approximation

"ME" - Traditional ME approach based on LDLT implementation

"OVUS" - One-variate univariate screening approach

"OVBS" - One-variate bivariate screening approach

"TGBME" - Trinh and Genz's bivariate conditioning approximation

"BME" - Bivariate ME approach

"TVBS" - Two-variate bivariate screening approach

INPUTS

\*\* mu = c( mu1,mu2,mu3,mu4 ) Kx1 (K>=2) vector of means (say K=4)

\*\* cov = matrix( c(cov11, cov12, cov13, cov14, cov is KxK, covariance matrix (if .covarr=1)

cov12, cov22, cov23, cov24,

cov13, cov23, cov33, cov34,

cov14, cov24, cov34, cov44), nrow=4)

\*\* cov = matrix( c(1 , rho12, rho13, rho14, cov is KxK, covariance matrix (if .covarr=1)

rho12, 1 , rho23, rho24,

rho13, rho23, 1 , rho34,

rho14, rho24, rho34, 1 ), nrow=4)

\*\* x = c( x1,x2,x3,x4 ); Kx1 vector of abscissae

\*\* s is a seed value that is relevant only for the SSJ method or for the randomized versions of other methods (when using .optimal=3). This argument is optional.

OUTPUTS

\*\* pdfmvnanalytic(x,mu,cov)[[1]] = Cumulative probability at x 1x1 scalar

\*\* pdfmvnanalytic(x,mu,cov)[[2]] = Gradient of distribution function with respect to mu

c( dP/dmu1, Kx1 vector

dP/dmu2,

dP/dmu3,

dP/dmu4 )

\*\* pdfmvnanalytic(x,mu,cov)[[3]] = Gradient of distribution function with respect to elements of covariance/correlation matrix

c( dP/dcov11 [K\*(K+1)/2 x 1] vector if .covar=1

dP/dcov12

dP/dcov13

dP/dcov14

dP/dcov22

dp/dcov23

dP/dcov24

dP/dcov33

dP/dcov34

dP/dcov44 )

c( dP/drho12 [K\*(K-1)/2 x 1] vector if .covar=0

dP/drho13

dP/drho14

dP/drho23

dP/drho24

dP/drho34 )

\*\* pdfmvnanalytic(x,mu,cov)[[4]] = Gradient of distribution function with respect to x

c( dP/dx1, Kx1 vector

dP/dx2,

dP/dx3,

dP/dx4 )

\*\* pdfmvnanalytic(x,mu,cov)[[5]] = Vector seed coming out of SSJ method or from random ordering of other methods (having this, and using this as seed for next call of an MVNCD evaluation is helpful for model estimation where multiple MVNCD evaluations have to be undertaken); if not SSJ method and not random ordering for other methods, s1 is the same as the input scalar seed s

1. **Running the codes**

First all the functions and variables in the file CDFandPDFmvna\_v2.1.R has to be loaded.

source("/My/File/Path/CDFandPDFmvna\_v2.1.R")

To test if you have installed things correctly and to run the code, the reader can use the following values for the vector of mean, covariance matrix, abscissae and seed.

mu = c(1, 1.9, 1.1, 2, 0.9)

V = matrix(c( 3, 0.5, -0.4, 0.3, -0.2,

0.5, 2, 0.2, -0.3, 0.4,

-0.4, 0.2, 3, 0.5, -0.4,

0.3, -0.3, 0.5, 2, 0.2,

-0.2, 0.4, -0.4, 0.2, 1), nrow = 5)

x = c(2, 1, 2, 2, 1)

s = 1

/\* For the TVBS Method \*/

.covarr=1 /\* Input is a covariance matrix \*/

.optimal=0 /\* The abscissae are arranged in ascending order prior to

MVNCD evaluation \*/

.method="TVBS" /\* Invoking the TVBS method \*/

/\* The actual call to the MVNCD function \*/

result = cdfmvnanalytic(mu, V, x, seed)

/\* The call to the gradient function \*/

result = pdfmvnanalytic(mu, V, x, seed)

The table on the next page shows the MVNCD values (=P from the cdfmvnanalytic procedure) that users should obtain using the different methods (all that the user needs to do is to appropriately change the global variables of .perms, .optimal, and .method). To reduce clutter, we do not provide the gradient outputs from the pdfmvnanalytic procedure.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **.optimal = 0** | **.optimal = 1** | **.optimal = 2** | **.optimal = 3** |
| BME | 0.0504 | 0.0504 | 0.0507 | 0.0503 |
| ME | 0.0504 | 0.0504 | 0.0506 | 0.0505 |
| OVBS | 0.0504 | 0.0504 | 0.0505 | 0.0504 |
| OVUS | 0.0504 | 0.0504 | 0.0506 | 0.0504 |
| SSJ (.perms = 20) | 0.0502 | 0.0502 | 0.0502 | 0.0502 |
| TG | 0.0517 | 0.0517 | 0.0519 | 0.0535 |
| TGBME | 0.0511 | 0.0511 | 0.0510 | 0.0526 |
| TVBS | 0.0504 | -- | 0.0506 | 0.0503 |