Package 'exageostatr'

August 20, 2020

SystemRequirements GNU Make, GNU CMake, GCC Compiler Suite (C and Fortran), nlopt (>= 2.4.2 http://ab-initio.mit.edu), lapack (https://github.com/xianyi/OpenBLAS/releases), lapacke (https://github.com/xianyi/OpenBLAS/releases), blas (https://github.com/xianyi/OpenBLAS/releases), cblas (https://github.com/xianyi/OpenBLAS/releases), hwloc (>=1.11.5 https://www.open-mpi.org), gsl (>= 2.4 https://ftp.gnu.org)

Version 0.1.0

Date 2020-08-20

Title R Package Demonstrates the R / C Language Interface for Exageostat

Author Sameh Abdulah <sameh.abdulah@kaust.edu.sa>

Maintainer Sameh Abdulah <sameh.abdulah@kaust.edu.sa>

Depends R (>= 2.0.1), assertthat (>= 0.2.1)

Description An R-wrapper for ExaGeoStat: a parallel high performance unified framework for geostatistics on manycore systems. Its abbreviation stands for Exascale Geostatistics. The framework aims at optimizing the likelihood function for a given spatial data to provide an efficient way to predict missing observations. The framework targets many-core systems: clusters of CPUs and GPUs.

License GPL (>= 2)

URL https://www.github.com/ecrc/exageostatr

OS_type unix

RoxygenNote 7.1.1

NeedsCompilation yes

R topics documented:

`dst_mle` *Maximum Likelihood Evaluation (MLE) using Diagonal Super-tile (DST) method*

Description

Maximum Likelihood Evaluation (MLE) using Diagonal Super-tile (DST) method

Usage

```
dst_mle(
  data = list(x, y, z),
  dst_thick,
  dmetric = c("euclidean", "great_circle"),
  optimization = list(clb = c(0.001, 0.001, 0.001),
  cub = c(5, 5, 5), tol = 1e-04,
  max_iters = 100)
)
```

Arguments

`data`	A list of x vector (x-dim), y vector (y-dim), and z observation vector
`dst_thick`	A number - Diagonal Super-Tile (DST) diagonal thick
`dmetric`	A string - distance metric - "euclidean" or "great_circle"
`optimization`	A list of opt lb (clb), opt ub (cub), tol, max_iters

Value

vector of three values (theta1, theta2, theta3)

Examples

```
seed = 0  ##Initial seed to generate XY locs.
sigma_sq = 1 ##Initial variance.
beta = 0.03 ##Initial smoothness.
nu = 0.5 ##Initial range.
dmetric = "euclidean" ##"euclidean" or "great_circle" distance.
n = 900 ## The number of locations (n must be a square number, n=m^2).
dst_thick = 3 ##Number of used Diagonal Super Tile (DST).
exageostat_init(hardware = list (ncores = 4, ngpus = 0,
                ts = 320, pgrid  = 1, qgrid  = 1)) ##Initiate exageostat instance
data = simulate_data_exact(sigma_sq, beta, nu, dmetric,
                           n, seed) ##Generate Z observation vector
##Estimate MLE parameters (TLR approximation)
result = dst_mle(data, dst_thick, dmetric, optimization = list(
                 clb = c(0.001, 0.001, 0.001),
                 cub = c(5, 5, 5), tol = 1e-4, max_iters = 4))
print(result)
exageostat_finalize() ##Finalize exageostat instance
```

exact_mle *Maximum Likelihood Evaluation using exact method*

Description

Maximum Likelihood Evaluation using exact method

Usage

```
exact_mle(
  data = list(x, y, z),
  dmetric = c("euclidean", "great_circle"),
  optimization = list(clb = c(0.001, 0.001, 0.001),
                      cub = c(5, 5, 5), tol = 1e-04,
                      max_iters = 100)
)
```

Arguments

data	A list of x vector (x-dim), y vector (y-dim), and z observation vector
dmetric	A string - distance metric - "euclidean" or "great_circle"
optimization	A list of opt lb values (clb), opt ub values (cub), tol, max_iters

Value

vector of three values (theta1, theta2, theta3)

Examples

```
seed = 0  ##Initial seed to generate XY locs.
sigma_sq = 1 ##Initial variance.
beta = 0.1 ##Initial smoothness.
nu = 0.5 ##Initial range.
dmetric = "euclidean" ##"euclidean" or "great_circle" distance.
n = 144 ## The number of locations (n must be a square number, n=m^2).
theta_out[1:3] = -1.99 ## Initial outputs
exageostat_init(hardware = list (ncores = 2, ngpus = 0,
                ts = 32, pgrid  = 1, qgrid  = 1)) ##Initiate exageostat instance
data = simulate_data_exact(sigma_sq, beta, nu, dmetric,
                           n, seed) ##Generate Z observation vector
##Estimate MLE parameters (Exact)
result = exact_mle(data, dmetric, optimization = list(
                  clb = c(0.001, 0.001, 0.001), cub = c(5, 5, 5),
                  tol = 1e-4, max_iters = 1))
print(result)
exageostat_finalize() ##Finalize exageostat instance
```

exageostat_finalize *Finalize the current instance of ExaGeoStatR*

Description

Finalize the current instance of ExaGeoStatR

Usage

```
exageostat_finalize()
```

Value

N/A

Examples

```
exageostat_finalize()
```

exageostat_init *Initial an instance of ExaGeoStatR*

Description

Initial an instance of ExaGeoStatR

Usage

```
exageostat_init(
  hardware = list(ncores = 2, ngpus = 0, ts = 320,
                  lts = 0, pgrid = 1, qgrid = 1)
)
```

Arguments

hardware	A list of ncores, ngpus, tile size, pgrid, and qgrid

Value

N/A

Examples

```
exageostat_init(hardware = list (ncores=2,  ngpus=0, ts=320,
                                 lts=0, pgrid=1,  qgrid=1 ))
exageostat_init(hardware = list (ncores=1,  ngpus=2, ts=320,
                                 lts=0, pgrid=1,  qgrid=1 ))
exageostat_init(hardware = list (ncores=26, ngpus=0, ts=320,
                                 lts=0, pgrid=3,  qgrid=4 ))
```

simulate_data_exact *Simulate Geospatial data (x, y, z)*

Description

Simulate Geospatial data (x, y, z)

Usage

```
simulate_data_exact(
  sigma_sq,
  beta,
  nu,
  dmetric = c("euclidean", "great_circle"),
  n,
  seed = 0
)
```

Arguments

sigma_sq	A number - variance parameter
beta	A number - smoothness parameter)
nu	A number - range parameter
dmetric	A string - distance metric - "euclidean" or "great_circle"
n	A number - data size
seed	A number - seed of random generation

Value

a list of of three vectors (x, y, z)

Examples

```
seed = 0  ##Initial seed to generate XY locs.
sigma_sq = 1 ##Initial variance.
beta = 0.1 ##Initial smoothness.
nu = 0.5 ##Initial range.
dmetric = "euclidean" ##"euclidean" or "great_circle" distance.
n = 1600 ## The number of locations (n must be a square number, n=m^2).
exageostat_init(hardware = list (ncores = 2, ngpus = 0,
                ts = 320, pgrid  = 1, qgrid  = 1)) ##Initiate exageostat instance
data = simulate_data_exact(sigma_sq, beta, nu, dmetric,
                           n, seed) ##Generate Z observation vector
data
exageostat_finalize() ##Finalize exageostat instance
```

`simulate_obs_exact` *Simulate Geospatial data given (x, y) locations*

Description

Simulate Geospatial data given (x, y) locations

Usage

```
simulate_obs_exact(
  x,
  y,
  sigma_sq,
  beta,
  nu,
  dmetric = c("euclidean", "great_circle")
)
```

Arguments

`x`	A vector (x-dim)
`y`	A vector (y-dim)
`sigma_sq`	A number - variance parameter
`beta`	A number - smoothness parameter)
`nu`	A number - range parameter
`dmetric`	A string - distance metric - "euclidean" or "great_circle"

Value

a list of of three vectors (x, y, z)

Examples

```
sigma_sq = 1 ##Initial variance.
beta = 0.1 ##Initial smoothness.
nu = 0.5 ##Initial range.
dmetric = "euclidean" ##"euclidean" or "great_circle" distance.
n = 1600 ## The number of locations (n must be a square number, n=m^2)
x = rnorm(n, 0, 1)     #x measurements of n locations.
y = rnorm(n, 0, 1)    #y measurements of n locations.
exageostat_init(hardware = list (ncores = 2, ngpus = 0, ts = 320,
                pgrid  = 1, qgrid  = 1)) ##Initiate exageostat instance
data = simulate_obs_exact(x, y, sigma_sq, beta, nu,
                 dmetric) ##Generate Z observation vector based on given locations
exageostat_finalize() ##Finalize exageostat instance
```

`tlr_mle` *Maximum Likelihood Evaluation (MLE) using Tile Low-Rank (TLR) method*

Description

Maximum Likelihood Evaluation (MLE) using Tile Low-Rank (TLR) method

Usage

```
tlr_mle(
  data = list(x, y, z),
  tlr_acc = 9,
  tlr_maxrank = 400,
  dmetric = c("euclidean", "great_circle"),
  optimization = list(clb = c(0.001, 0.001, 0.001),
                      cub = c(5, 5, 5), tol = 1e-04,
                      max_iters = 100)
)
```

Arguments

`data`	A list of x vector (x-dim), y vector (y-dim), and z observation vector
`tlr_acc`	A number - TLR accuracy level
`tlr_maxrank`	A string - TLR max rank
`dmetric`	A string - distance metric - "euclidean" or "great_circle"
`optimization`	A list of opt lb values (clb), opt ub values (cub), tol, max_iters

Value

vector of three values (theta1, theta2, theta3)

Examples

```
seed = 0  ##Initial seed to generate XY locs.
sigma_sq = 1 ##Initial variance.
beta = 0.03 ##Initial smoothness.
nu = 0.5 ##Initial range.
dmetric = "euclidean" ##"euclidean" or "great_circle" distance.
n = 900 ## The number of locations (n must be a square number, n=m^2).
tlr_acc = 7 ##Approximation accuracy 10^-(acc)
tlr_maxrank = 150 ##Max Rank
exageostat_init(hardware = list (ncores = 2, ngpus = 0, ts = 320,
                lts = 600, pgrid  = 1, qgrid  = 1)) ##Initiate exageostat instance
data = simulate_data_exact(sigma_sq, beta, nu, dmetric,
                           n, seed) ##Generate Z observation vector
##Estimate MLE parameters (TLR approximation)
result = tlr_mle(data, tlr_acc, tlr_maxrank, dmetric, optimization =
                 list(clb = c(0.001, 0.001, 0.001), cub = c(5, 5, 5),
                      tol = 1e-4, max_iters = 4))
print(result)
exageostat_finalize() ##Finalize exageostat instance
```

Index