Package 'KRIG'

January 30, 2018

Type Package

Title Spatial Statistic with Kriging

Version 0.1.0

Encoding UTF-8

Date 2018-01-27

License LGPL-3

Description Implements different methods for spatial statistics, in particular focused in Kriging based models. We count with different implemented models, simple, ordinary and universal forms of Kriging, co-Kriging and regression Kriging models. Includes, multivariate sensitivity analysis under an approximation designed over reproducing kernel Hilbert spaces and computation of Sobol indexes under this framework.

URL https://github.com/pedroguarderas/KRIG

BugReports https://github.com/pedroguarderas/KRIG/issues

Depends R (>= 3.3)

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 6.0.1

Imports Rcpp, Rdpack

Suggests data.table, colorRamps, plotly, knitr, rmarkdown, testthat

LazyData true

VignetteBuilder knitr

RdMacros Rdpack

NeedsCompilation yes

Author Pedro Guarderas [aut, cre],
Daniel Lagos [ctb],
Andrés Lopez [ctb]

Maintainer Pedro Guarderas <pedro.felipe.guarderas@gmail.com>

Repository CRAN

Date/Publication 2018-01-30 18:50:00 UTC

R topics documented:

`KRIG-package` *Spatial Statistics with Kriging*

Description

Implements different methods for spatial statistics, in particular focused with Kriging based models. We count with different implemented models, simple, ordinary and universal forms of Kriging, co-Kriging and regression Kriging models. Includes, multivariate sensitivity analysis under an approximation designed over reproducing kernel Hilbert spaces and computation of Sobol indexes under this framework.

The linear algebra operations are supported by RcppArmadillo.

Author(s)

Maintainer: Pedro Guarderas <pedro.felipe.guarderas@gmail.com>

Other contributors:

- Daniel Lagos [contributor]
- Andrés Lopez [contributor]

References

Matheron G (1970). *La Théorie des variables régionalisées, et ses applications*, series Les Cahiers du Centre de morphologie mathématique de Fontainebleau. École Nationale Supérieure des Mines de Paris.

Chilès, J.P. and Delfiner, P. (2012). *Geostatistics: Modeling Spatial Uncertainty*, series Wiley Series in Probability and Statistics. Wiley. ISBN 978-0-470-18315-1.

Durrande, N., Ginsbourger, D., Roustant, O. and Carraro, L. (2013). "ANOVA kernels and RKHS of zero mean functions for model-based sensitivity analysis." *Journal of Multivariate Analysis*, **115**(C), pp. 57-67.

Rasmussen CE and Williams CKI (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press. ISBN 026218253X.

Stein M (1999). *Interpolation of Spatial Data, Some Theory of Kriging*, series Springer Series in Satatistics. Springer. ISBN 978-1-4612-7166-6.

Clark I (1979). *Practical Geostatistics*. Applied Science Pub.

Aronszajn, N. (1950). "Theory of reproducing kernels." *Transactions of the American Mathematical Society*, **68**(3), pp. 337–404. http://dx.doi.org/10.2307/1990404.

See Also

Useful links:

- https://github.com/pedroguarderas/KRIG
- Report bugs at https://github.com/pedroguarderas/KRIG/issues

Examples

```
library( KRIG )
vignette( topic = 'copper_mining_2d', package = 'KRIG' )
```

Copper *Copper mining data*

Description

This is reproduced from the original description for the dataset. A simulation based on a stockpile of mined material in the former Soviet Union. Boreholes have been drilled into the dump. The drill core is cut every 5 meters and assayed for copper and cobalt content in percentage by weight. This is the only three dimensional set of tutorial data. Coordinates are in meters.

Usage

```
Copper
```

Format

A data table with 442 and 7 columns.

a cod id

s sample id

x1 first coordinate

x2 second coordinate

x3 third coordinate

Z copper grade

C value

Source

www.edumine.com.

References

Dr. Isobel Clark

Examples

```
data( 'Copper', packages = 'KRIG' )
```

---

exp_kernel *Exponential kernel.*

Description

Isotropic kernel.

Usage

```
exp_kernel(h, sigma = 1, theta = 1)
```

Arguments

h	distance variable.
sigma	amplitude parameter.
theta	smoothness parameter.

Value

Real value.

Author(s)

Pedro Guarderas <pedro.felipe.guarderas@gmail.com>.

Examples

```
h<-0.3
sigma<-1.0
theta<-10
exp_kernel( h, sigma, theta )
```

`gaussian_kernel` *Gaussian kernel.*

Description

Isotropic kernel.

Usage

```
gaussian_kernel(h, sigma = 1, theta = 1)
```

Arguments

h	distance variable.
sigma	amplitude parameter.
theta	smoothness parameter.

Value

Real value.

Author(s)

Pedro Guarderas <pedro.felipe.guarderas@gmail.com>.

Examples

```
h<-4
alpha<-2
gaussian_kernel( h, alpha )
```

integrate_kernel *Complete kernel integral.*

Description

This function is part of the routines employed in the sensitivity analysis, calculate the integral in both coordinate x and y of the kernel, over the square domain give by the limits a and b.

Usage

```
integrate_kernel(Kern, a, b, n)
```

Arguments

Kern	Kernel function.
a	Inferior limit for the integral in each coordinate.
b	Superior limit for the integral in each coordinate.
n	Number of uniform division to compute the integral.

Value

Real value with the integral value.

Author(s)

Pedro Guarderas <pedro.felipe.guarderas@gmail.com>.

See Also

For a complete application you can check the documentation of Krigvar.

Kanova *KANOVA, kernel anova under RKHS approximations.*

Description

Under an approximation to the sensitivity analysis based in variance computation the different indexes of combinatorial sensitivity values can be computed employing the values of kernel integrals.

Usage

```
Kanova(Kernels, Integral, X)
```

Arguments

Kernels
: data.frame of kernels composed by four columns.

1. Kernel name.
2. Inferior limit for integral.
3. Superior limit for integral.
4. Number of steps for discretization of integrals.

Integral
: A list containing the results of kernel integrals of the functions `vector_integrate_kernel`.

X
: matrix containing in each row the coordinate where the one coordinate integrals will be evaluated.

Value

List with containing the Gamma 3D array where the different combination variance are stocked and the total matrix variance named Kanova.

Author(s)

Pedro Guarderas <pedro.felipe.guarderas@gmail.com>.

References

Durrande, N., Ginsbourger, D., Roustant, O. and Carraro, L. (2013). "ANOVA kernels and RKHS of zero mean functions for model-based sensitivity analysis." *Journal of Multivariate Analysis*, **115**(C), pp. 57-67. Aronszajn, N. (1950). "Theory of reproducing kernels." *Transactions of the American Mathematical Society*, **68**(3), pp. 337–404. http://dx.doi.org/10.2307/1990404.

See Also

For a complete application you can check the documentation of `Krigvar`.

Kov — *Spatial covariance matrix.*

Description

To compute a kriging, it is necessary the spatial covariance matrix. The spatial covariance could computed between to sets of points X and Y with different dimension and the result it is not necessarily a square matrix.

Usage

```
Kov(X, Y, Kern, symmetric = FALSE)
```

Arguments

`X`	First set of spatial points.
`Y`	Second set of spatial points.
`Kern`	Kernel function.
`symmetric`	If result of computation will be a square matrix, the time computation can be improved setting this parameter to TRUE, the default is FALSE.

Value

The spatial covariance matrix.

Author(s)

Pedro Guarderas <`pedro.felipe.guarderas@gmail.com`>.

See Also

For a complete application you can check the documentation of `Krig`.

`Krig` *Kriging computation.*

Description

Computes the kriging linear estimator for different types of kriging models.

Usage

```
Krig(Z, K, k, G, g, type = "ordinary", cinv = "syminv")
```

Arguments

`Z`	Observed values of the spatial process.
`K`	Covariance matrix computed for the position X where the spatial process Z was observed.
`k`	Covariance matrix computed for the position X where the spatial process Z was observed and the position Y where the spatial process Z will be predicted.
`G`	When universal kriging will be computed, this matrix represents the values of the of the functions representing the mean of the process Z, evaluated in the spatial points X where the spatial process was first observed.
`g`	When universal kriging will be computed, this matrix represents the evaluation of the functions representing the mean over the new position points Y where the spatial process Z will be predicted.
`type`	Type of kriging model, possible values are: simple, ordinary, universal.

cinv — Specifies how the inverse of the covariance matrix K will be computed. Possible values are: syminv = symmetric matrix inverse computation, inv = usual armadillo inverse computation, cholinv = Cholesky based inverse computation, ginv = given inverse not necessary to compute inverse at all.

Value

Depending of the type of analysis the list of results change.

Z — New estimated values for Z.

L — Linear coefficients determined by kriging.

J — Inverse of the covariance matrix.

tau — Factor computed in the ordinary and universal kriging.

alpha — Factor computed in the ordinary kriging.

A — Factor computed in the universal kriging.

Author(s)

Pedro Guarderas <pedro.felipe.guarderas@gmail.com>.

Examples

```
library( KRIG )
vignette( topic = 'simple_kriging', package = 'KRIG' )
vignette( topic = 'ordinary_kriging', package = 'KRIG' )
vignette( topic = 'universal_kriging', package = 'KRIG' )
vignette( topic = 'copper_mining_2d', package = 'KRIG' )
```

Krigidx — *Combinatorial variance computation.*

Description

For a given combination this function computes the associated variance for the variable enumerated by the combination values.

Usage

```
Krigidx(KF, comb, X, Gamma)
```

Arguments

KF — values of the kernel integral evaluations.

comb — Combination.

X — Points in the grid.

Gamma — Cube with integral results.

Value

Real value of sensitivity.

Author(s)

Pedro Guarderas <pedro.felipe.guarderas@gmail.com>.

References

Durrande, N., Ginsbourger, D., Roustant, O. and Carraro, L. (2013). "ANOVA kernels and RKHS of zero mean functions for model-based sensitivity analysis." *Journal of Multivariate Analysis*, **115**(C), pp. 57-67. Aronszajn, N. (1950). "Theory of reproducing kernels." *Transactions of the American Mathematical Society*, **68**(3), pp. 337–404. http://dx.doi.org/10.2307/1990404.

See Also

For a complete application you can check the documentation of `Krigvar`.

`Krigvar` *Combinatorial variance computation.*

Description

Computation of variance

Usage

```
Krigvar(KF, Gamma)
```

Arguments

`KF`	values of the kernel integral evaluations.
`Gamma`	Cube with integral results.

Value

Real value of sensitivity.

Author(s)

Pedro Guarderas <pedro.felipe.guarderas@gmail.com>.

References

Durrande, N., Ginsbourger, D., Roustant, O. and Carraro, L. (2013). "ANOVA kernels and RKHS of zero mean functions for model-based sensitivity analysis." *Journal of Multivariate Analysis*, **115**(C), pp. 57-67. Aronszajn, N. (1950). "Theory of reproducing kernels." *Transactions of the American Mathematical Society*, **68**(3), pp. 337–404. http://dx.doi.org/10.2307/1990404.

Examples

```
library( KRIG )
options( stringsAsFactors = FALSE )

kernel_1<-function( x, y ) exp( -0.5*(x-y)^2)
kernel_2<-function( x, y ) exp( -0.7*(x-y)^2)
kernel_3<-function( x, y ) exp( -0.1*(x-y)^2)

Kernels<-data.frame( kernel = c( 'kernel_1', 'kernel_2', 'kernel_3' ),
                     min = c( -1, -1, -2 ),
                     max = c( 1, 1, 2 ),
                     n = c( 100, 100, 100 ) )

n<-20
X<-matrix( c( seq( -1, 1, length.out = n ),
              seq( -1, 1, length.out = n ),
              seq( -2, 2, length.out = n ) ), n, 3 )

KI<-list_integrate_kernel( Kernels, X )
GK<-Kanova( Kernels, KI, X )

f<-function( x ) x[1] + 30 * x[2] + 60 * x[3]
Func<-apply( X, 1, FUN = f )

KF<-solve( GK$Kanova + diag( 1e-8, n, n ), Func )

SbI<-NULL
for ( j in 1:3 ) {
  CB<-combn( 1:3, j )
  for ( l in 1:ncol( CB ) ) {
    SbI<-c( SbI, Krigidx( KF, CB[,l], X, GK$Gamma ) )
    names(SbI)[length(SbI)]<-paste( 'C.', paste( CB[,l], collapse='.' ), sep = '' )
  }
}

Var<-Krigvar( KF, GK$Gamma )

SVar<-sum( SbI / Var )
```

`linear_kernel` *Linear kernel*

Description

Anisotropic kernel defined by the scalar product.

Usage

```
linear_kernel(x, y, alpha)
```

Arguments

x	first column vector.
y	second column vector.
alpha	amplitude parameter.

Value

Real value.

Author(s)

Pedro Guarderas <pedro.felipe.guarderas@gmail.com>.

Examples

```
n<-10
x<-matrix( runif( n ), n, 1 )
y<-matrix( runif( n ), n, 1 )
alpha<-0.5
linear_kernel( x, y, alpha )
```

---

list_integrate_kernel *Integrals of a list of kernels.*

Description

This function is part of the routines employed in the sensitivity analysis, computes vector of integrals and complete integrals of kernels specified in the data frame Kernels.

Usage

```
list_integrate_kernel(Kernels, X)
```

Arguments

`Kernels` data.frame of kernels composed by four columns.

1. Kernel name.
2. Inferior limit for integral.
3. Superior limit for integral.
4. Number of steps for discretization of integrals.

`X` matrix containing in each row the coordinate where the one coordinate integrals will be evaluated.

Value

List with one coordinate integrals and complete kernel integrals.

Author(s)

Pedro Guarderas <pedro.felipe.guarderas@gmail.com>.

See Also

For a complete application you can check the documentation of Krigvar.

---

matern_kernel *Matérn kernel.*

Description

Isotropic kernel.

Usage

```
matern_kernel(h, v = 2, sigma = 1, theta = 1)
```

Arguments

h	distance variable.
v	power parameter.
sigma	amplitude parameter.
theta	smoothness parameter.

Value

Real value.

Author(s)

Pedro Guarderas <pedro.felipe.guarderas@gmail.com>.

Examples

```
h<-4.0
v<-2.0
sigma<-2.0
theta<-100.0
matern_kernel( h, v, sigma, theta )
```

multilog_kernel *Multilog kernel.*

Description

Isotropic kernel.

Usage

```
multilog_kernel(h, R = 1)
```

Arguments

h	distance parameter.
R	displacement parameter.

Value

Real value.

Author(s)

Pedro Guarderas <pedro.felipe.guarderas@gmail.com>.

Examples

```
h<-4
R<-2.3
multilog_kernel( h, R )
```

nat_cubic_spline_kernel *Natural cubic spline kernel.*

Description

Isotropic kernel.

Usage

```
nat_cubic_spline_kernel(h, R = 1)
```

Arguments

h	distance variable.
R	displacement parameter.

Value

Real value.

Author(s)

Pedro Guarderas <pedro.felipe.guarderas@gmail.com>.

Examples

```
h<-4.0
R<-2.3
nat_cubic_spline_kernel( h, R )
```

`polynomial_kernel` *Polynomial kernel*

Description

Anisotropic kernel defined like a polynomial in the scalar product.

Usage

```
polynomial_kernel(x, y, alpha, beta, n)
```

Arguments

x	first column vector.
y	second column vector.
alpha	amplitude parameter.
beta	displacement parameter.
n	power parameter.

Value

Real value.

Author(s)

Pedro Guarderas <pedro.felipe.guarderas@gmail.com>. N<-10 x<-matrix(runif(N), N, 1) y<-matrix(runif(N), N, 1) alpha<-0.5 beta<-2 n<-3 polynomial_kernel(x, y, alpha, beta, n)

spherical_kernel *Spherical kernel.*

Description

Isotropic kernel commonly employed in geostatistic.

Usage

```
spherical_kernel(h, phi, theta)
```

Arguments

h	distance variable.
phi	amplitude parameter.
theta	smoothness parameter.

Value

Real value.

Author(s)

Pedro Guarderas <pedro.felipe.guarderas@gmail.com>.

Examples

```
h<-4.0
phi<-1.0
theta<-100.0
spherical_kernel( h, phi, theta )
```

square_kernel *Square kernel.*

Description

Isotropic kernel given by the square distance.

Usage

```
square_kernel(h, alpha = 1)
```

Arguments

h	distance variable.
alpha	amplitude parameter.

Value

Real value.

Author(s)

Pedro Guarderas <pedro.felipe.guarderas@gmail.com>.

Examples

```
h<-4
alpha<-2
square_kernel( h, alpha )
```

---

thin_plate_kernel *Thin plate kernel.*

Description

Isotropic kernel.

Usage

```
thin_plate_kernel(h, R = 1)
```

Arguments

h	distance variable.
R	displacement parameter.

Value

Real value.

Author(s)

Pedro Guarderas <pedro.felipe.guarderas@gmail.com>.

Examples

```
h<-4.0
R<-0.5
thin_plate_kernel( h, R )
```

triangular_kernel *Triangular kernel.*

Description

Isotropic kernel defined with the max function.

Usage

```
triangular_kernel(h, c = 1, alpha = 1)
```

Arguments

h	distance variable.
c	amplitude parameter.
alpha	maximum distance value.

Value

Real value.

Author(s)

Pedro Guarderas <pedro.felipe.guarderas@gmail.com>.

Examples

```
h<-0.2
c<-5
alpha<-2
triangular_kernel( h, c, alpha )
```

variogram *Computes the variogram.*

Description

This useful function is commonly employed in the study of isotropic stationary spatial processes.

Usage

```
variogram(Z, X, d)
```

Arguments

Z — Vector of observations.

X — Points matrix.

d — Distance function.

Value

Variogram vector.

Author(s)

Pedro Guarderas <pedro.felipe.guarderas@gmail.com>.

Examples

```
library( KRIG )
vignette( topic = 'copper_mining_2d', package = 'KRIG' )
```

---

vector_integrate_kernel

One coordinate kernel integral.

Description

This function is part of the routines employed in the sensitivity analysis, takes the kernel k and for each fixed coordinate in x, the integral in the second variable y, is computed in the interval a to b by taking n uniform steps.

Usage

```
vector_integrate_kernel(Kern, x, a, b, n)
```

Arguments

Kern — Kernel function.

x — Column vector with values for the first coordinate of the kernel.

a — Inferior limit for the integral in y.

b — Superior limit for the integral in y.

n — Number of uniform division to compute the integral.

Value

Vector with integrals while the x coordinate is fixed.

Author(s)

Pedro Guarderas <pedro.felipe.guarderas@gmail.com>.

See Also

For a complete application you can check the documentation of Krigvar.

---

weight_pow_dist *Generic weighted p-distance*

Description

Many isotropic models can be defined employing a distance. In particular this function implements a distance employing weights and different powers.

Usage

```
weight_pow_dist(x, y, w, p)
```

Arguments

x	First vector.
y	Second vector.
w	Weights for every coordinate in the vectors.
p	Powers for every coordinate in the vectors. To define a true norm every coordinate has to be greater than 1.

Value

Real value of the weighted p-distance.

Author(s)

Pedro Guarderas <pedro.felipe.guarderas@gmail.com>.

Examples

```
library(KRIG)

n<-1e5
x<-runif( n )
y<-runif( n )
p<-rep(2.5,n)
w<-runif(n)

kw<-weight_pow_dist( x, y, w, p )
nw<-sum( w * abs( x - y )^p )

nw == kw
```

Index