Package ‘rlibkriging’

January 13, 2021

Type Package

Title Kriging model through libKriging binding

Version 0.1-10

Date 2020-12-21

Author Pascal Havé <`hpwxf@haveneer.com`>, Yann Richet <`yann.richet@irsn.fr`>

Maintainer Pascal Havé <`hpwxf@haveneer.com`>

Description
Binding libKriging to R, and provide DiceKriging features with improved performance.

License Apache License (¿= 2)

Encoding UTF-8

LinkingTo Rcpp, RcppArmadillo

Depends R (¿= 2.14)

Imports Rcpp (¿= 0.12.11), methods

Suggests testthat, DiceKriging, utils

SystemRequirements GNU make

URL `https://github.com/libKriging`

RoxygenNote 7.1.1

R topics documented:

`as.list.Kriging` *List Kriging object content*

Description

List Kriging object content

Usage

```
## S3 method for class 'Kriging'
as.list(x, ...)
```

Arguments

`x`	S3 Kriging object
`...`	Ignored

Value

list of Kriging object fields: kernel, optim, objective, theta, sigma2, X, centerX, scaleX, y, centerY, scaleY, regmodel, F, T, M, z, beta

Author(s)

Yann Richet (yann.richet@irsn.fr)

Examples

```
f = function(x) 1-1/2*(sin(12*x)/(1+x)+2*cos(7*x)*x^5+0.7)
set.seed(123)
X <- as.matrix(runif(5))
y <- f(X)
r <- Kriging(y, X, "gauss")
l = as.list(r)
cat(paste0(names(l)," =" ,l,collapse="\n"))
```

as_km — *Build a "as_km" object, which extends DiceKriging::km S4 class.*

Description

Build a "as_km" object, which extends DiceKriging::km S4 class.

Usage

```
as_km(...)
```

Arguments

`...`	args

Value

as_km/km object

Author(s)

Yann Richet (yann.richet@irsn.fr)

as_km.default — *Build a DiceKriging "km" like object.*

Description

Build a DiceKriging "km" like object.

Usage

```
## Default S3 method:
as_km(
  formula = ~1,
  design,
  response,
  covtype = "matern5_2",
  coef.cov = NULL,
  coef.var = NULL,
  coef.trend = NULL,
  estim.method = "MLE",
  optim.method = "BFGS",
  parinit = NULL,
  ...
)
```

Arguments

`formula`	R formula object to setup the linear trend (aka Universal Kriging). Supports ~1, ~. and ~.^2
`design`	data.frame of design of experiments
`response`	array of output values
`covtype`	covariance structure. Supports "gauss", "exp", ...
`coef.cov`	fixed covariance range value (so will not optimize if given)
`coef.var`	fixed variance value (so will not estimate if given)
`coef.trend`	fixed trend value (so will not estimate if given)
`estim.method`	estimation criterion. Supports "MLE" or "LOO"
`optim.method`	optimization algorithm used on estim.method objective. Supports "BFGS"
`parinit`	initial values of covariance range which will be optimzed using optim.method
`...`	Ignored

Value

as_km object, extends DiceKriging::km (plus contains a "Kriging" field which contains original object)

Author(s)

Yann Richet (yann.richet@irsn.fr)

Examples

```
# a 16-points factorial design, and the corresponding response
d <- 2; n <- 16
design.fact <- expand.grid(x1=seq(0,1,length=4), x2=seq(0,1,length=4))
y <- apply(design.fact, 1, DiceKriging::branin)

#library(DiceKriging)
# kriging model 1 : matern5_2 covariance structure, no trend, no nugget effect
#m1 <- km(design=design.fact, response=y,covtype = "gauss",parinit = c(.5,1),control = list(trace=F))
as_m1 <- as_km(design=design.fact, response=y,covtype = "gauss",parinit = c(.5,1))
```

`as_km.Kriging` *Convert a "Kriging" object to a DiceKriging::km one.*

Description

Convert a "Kriging" object to a DiceKriging::km one.

Usage

```
## S3 method for class 'Kriging'
as_km(k, .call = NULL)
```

Arguments

`k`	”Kriging” object
`.call`	Force the ”call” filed in km object

Value

as_km object, extends DiceKriging::km plus contains ”Kriging” field

Author(s)

Yann Richet (yann.richet@irsn.fr)

Examples

```
f = function(x) 1-1/2*(sin(12*x)/(1+x)+2*cos(7*x)*x^5+0.7)
set.seed(123)
X <- as.matrix(runif(5))
y <- f(X)
r <- Kriging(y, X, "gauss")
print(r)
k <- as_km(r)
print(k)
```

`Kriging` *Build a ”Kriging” object from libKriging.*

Description

Build a ”Kriging” object from libKriging.

Usage

```
Kriging(
  y,
  X,
  kernel,
  regmodel = "constant",
  normalize = FALSE,
  optim = "BFGS",
  objective = "LL",
  parameters = NULL
)
```

Arguments

`y`	Array of response values
`X`	Matrix of input design
`kernel`	Covariance model: "gauss", "exp", ...
`regmodel`	Universal Kriging linear trend: "constant", "linear", "interactive" ("constant" by default)
`normalize`	Normalize X and y in [0,1] (FALSE by default)
`optim`	Optimization method to fit hyper-parameters: "BFGS", "Newton" (uses objective Hessian), "none" (keep initial "parameters" values)
`objective`	Objective function to optimize: "LL" (log-Likelihood, by default), "LOO" (leave one out)
`parameters`	Initial hyper parameters: list(sigma2=..., theta=...). If theta has many rows, each is used as a starting point for optim.

Value

S3 Kriging object. Should be used with its predict, simulate, update methods.

Author(s)

Yann Richet (yann.richet@irsn.fr)

`leaveOneOut` *Compute model leave-One-Out error at given args*

Description

Compute model leave-One-Out error at given args

Usage

```
leaveOneOut(...)
```

Arguments

`...`	args

Value

leave-One-Out

`leaveOneOut.Kriging` *Compute leave-One-Out of Kriging model*

Description

Compute leave-One-Out of Kriging model

Usage

```
## S3 method for class 'Kriging'
leaveOneOut(object, theta, grad = FALSE)
```

Arguments

`object`	S3 Kriging object
`theta`	new points in model output space
`grad`	return Gradient ? (default is TRUE)

Value

leave-One-Out computed for given theta

Author(s)

Yann Richet (yann.richet@irsn.fr)

Examples

```
f = function(x) 1-1/2*(sin(12*x)/(1+x)+2*cos(7*x)*x^5+0.7)
set.seed(123)
X <- as.matrix(runif(5))
y <- f(X)
r <- Kriging(y, X, "gauss",objective="LOO")
print(r)
loo = function(theta) leaveOneOut(r,theta)$leaveOneOut
t = seq(0.0001,2,,101)
  plot(t,loo(t),type='l')
  abline(v=as.list(r)$theta,col='blue')
```

`logLikelihood` *Compute model log-Likelihood at given args*

Description

Compute model log-Likelihood at given args

Usage

```
logLikelihood(...)
```

Arguments

`...`	args

Value

log-Likelihood

`logLikelihood.Kriging` *Compute log-Likelihood of Kriging model*

Description

Compute log-Likelihood of Kriging model

Usage

```
## S3 method for class 'Kriging'
logLikelihood(object, theta, grad = FALSE, hess = FALSE)
```

Arguments

`object`	S3 Kriging object
`theta`	new points in model output space
`grad`	return Gradient ? (default is TRUE)
`hess`	return Hessian ? (default is FALSe)

Value

log-Likelihood computed for given theta

Author(s)

Yann Richet (yann.richet@irsn.fr)

Examples

```
f = function(x) 1-1/2*(sin(12*x)/(1+x)+2*cos(7*x)*x^5+0.7)
set.seed(123)
X <- as.matrix(runif(5))
y <- f(X)
r <- Kriging(y, X, "gauss")
print(r)
ll = function(theta) logLikelihood(r,theta)$logLikelihood
t = seq(0.0001,2,,101)
  plot(t,ll(t),type='l')
  abline(v=as.list(r)$theta,col='blue')
```

`predict.as_km` *Overload DiceKriging::predict.km for as_km objects (expected faster).*

Description

Overload DiceKriging::predict.km for as_km objects (expected faster).

Usage

```
## S3 method for class 'as_km'
predict(
  object,
  newdata,
  type = "UK",
  se.compute = TRUE,
  cov.compute = FALSE,
  light.return = TRUE,
  bias.correct = FALSE,
  checkNames = FALSE,
  ...
)
```

Arguments

`object`	as_km object
`newdata`	matrix of points where to perform prediction
`type`	kriging family (”UK”)
`se.compute`	compute standard error (TRUE by default)
`cov.compute`	compute covariance matrix between newdata points (FALSE by default)
`light.return`	return no other intermediate objects (like T matrix) (default is TRUE)
`bias.correct`	fix UK variance and covaariance (defualt is FALSE)
`checkNames`	check consistency between object design data: X and newdata (default is FALSE)
`...`	Ignored

Value

list of predict data: mean, sd, trend, cov, upper95 and lower95 quantiles.

Author(s)

Yann Richet (yann.richet@irsn.fr)

Examples

```
# a 16-points factorial design, and the corresponding response
d <- 2; n <- 16
design.fact <- expand.grid(x1=seq(0,1,length=4), x2=seq(0,1,length=4))
y <- apply(design.fact, 1, DiceKriging::branin)

#library(DiceKriging)
# kriging model 1 : matern5_2 covariance structure, no trend, no nugget effect
#m1 <-      km(design=design.fact, response=y,covtype = "gauss",parinit = c(.5,1),control = list(trac
as_m1 <- as_km(design=design.fact, response=y,covtype = "gauss",parinit = c(.5,1))
as_p = predict(as_m1,newdata=matrix(.5,ncol=2),type="UK",checkNames=FALSE,light.return=TRUE)
```

`predict.Kriging` *Predict Kriging model at given points*

Description

Predict Kriging model at given points

Usage

```
## S3 method for class 'Kriging'
predict(object, x, stdev = T, cov = F, ...)
```

Arguments

`object`	S3 Kriging object
`x`	points in model input space where to predict
`stdev`	return also standard deviation (default TRUE)
`cov`	return covariance matrix between x points (default FALSE)
`...`	Ignored

Value

list containing: mean, stdev, cov

Author(s)

Yann Richet (yann.richet@irsn.fr)

Examples

```
f = function(x) 1-1/2*(sin(12*x)/(1+x)+2*cos(7*x)*x^5+0.7)
  plot(f)
set.seed(123)
X <- as.matrix(runif(5))
y <- f(X)
  points(X,y,col='blue')
r <- Kriging(y, X, "gauss")
x = seq(0,1,,101)
p_x = predict(r, x)
  lines(x,p_x$mean,col='blue')
  lines(x,p_x$mean-2*p_x$stdev,col='blue')
  lines(x,p_x$mean+2*p_x$stdev,col='blue')
```

`print.Kriging` *Print Kriging object content*

Description

Print Kriging object content

Usage

```
## S3 method for class 'Kriging'
print(x, ...)
```

Arguments

`x`	S3 Kriging object
`...`	Ignored

Author(s)

Yann Richet (yann.richet@irsn.fr)

Examples

```
f = function(x) 1-1/2*(sin(12*x)/(1+x)+2*cos(7*x)*x^5+0.7)
set.seed(123)
X <- as.matrix(runif(5))
y <- f(X)
r <- Kriging(y, X, "gauss")
print(r)
```

`simulate.as_km` *Overload DiceKriging::simulate.km for as_km objects (expected faster).*

Description

Overload DiceKriging::simulate.km for as_km objects (expected faster).

Usage

```
## S3 method for class 'as_km'
simulate(
  object,
  nsim = 1,
  seed = NULL,
  newdata,
  cond = TRUE,
  nugget.sim = 0,
  checkNames = FALSE,
  ...
)
```

Arguments

`object`	as_km object
`nsim`	number of response vector to simulate
`seed`	random seed
`newdata`	matrix of points where to perform prediction
`cond`	simulate conditional samples (only TRUE accepted)
`nugget.sim`	numercial ngget ,effect to avoid numerical unstabilities
`checkNames`	check consistency between object design data: X and newdata (default is FALSE)
`...`	Ignored

Value

length(x) x nsim matrix containing simulated path at newdata points

Author(s)

Yann Richet (yann.richet@irsn.fr)

Examples

```
f = function(x) 1-1/2*(sin(12*x)/(1+x)+2*cos(7*x)*x^5+0.7)
  plot(f)
set.seed(123)
X <- as.matrix(runif(5))
y <- f(X)
  points(X,y,col='blue')
k <- as_km(design=X, response=y,covtype = "gauss")
x = seq(0,1,,101)
s_x = simulate(k, nsim=3, newdata=x)
  lines(x,s_x[,1],col='blue')
  lines(x,s_x[,2],col='blue')
  lines(x,s_x[,3],col='blue')
```

`simulate.Kriging` *Simulate (conditional) Kriging model at given points*

Description

Simulate (conditional) Kriging model at given points

Usage

```
## S3 method for class 'Kriging'
simulate(object, nsim = 1, seed = 123, x, ...)
```

Arguments

`object`	S3 Kriging object
`nsim`	number of simulations to perform
`seed`	random seed used
`x`	points in model input space where to simulate
`...`	Ignored

Value

length(x) x nsim matrix containing simulated path at x points

Author(s)

Yann Richet (yann.richet@irsn.fr)

Examples

```
f = function(x) 1-1/2*(sin(12*x)/(1+x)+2*cos(7*x)*x^5+0.7)
  plot(f)
set.seed(123)
X <- as.matrix(runif(5))
y <- f(X)
  points(X,y,col='blue')
r <- Kriging(y, X, "gauss")
x = seq(0,1,,101)
s_x = simulate(r, nsim=3, x=x)
  lines(x,s_x[,1],col='blue')
  lines(x,s_x[,2],col='blue')
  lines(x,s_x[,3],col='blue')
```

---

update.as_km *Overload DiceKriging::update.km methd for as_km objects (expected faster).*

Description

Overload DiceKriging::update.km methd for as_km objects (expected faster).

Usage

```
## S3 method for class 'as_km'
update(
  object,
  newX,
  newy,
  newX.alreadyExist = FALSE,
  cov.reestim = TRUE,
  trend.reestim = cov.reestim,
  nugget.reestim = FALSE,
  newnoise.var = NULL,
  kmcontrol = NULL,
  newF = NULL,
  ...
)
```

Arguments

`object`	as_km object
`newX`	new design points: matrix of object@d columns
`newy`	new response points
`newX.alreadyExist`	if TRUE, newX contains some ppoints already in object@X
`cov.reestim`	fit object to newdata: estimate theta (only supports TRUE)

`trend.reestim`	fit object to newdata: estimate beta (only supports TRUE)
`nugget.reestim`	fit object to newdata: estimate nugget effect (only support FALSE)
`newnoise.var`	add noise to newy response
`kmcontrol`	parametrize fit (unsupported)
`newF`	
`...`	Ignored

Author(s)

Yann Richet (yann.richet@irsn.fr)

Examples

```
f = function(x) 1-1/2*(sin(12*x)/(1+x)+2*cos(7*x)*x^5+0.7)
  plot(f)
set.seed(123)
X <- as.matrix(runif(5))
y <- f(X)
  points(X,y,col='blue')
k <- as_km(design=X, response=y,covtype = "gauss")
x = seq(0,1,,101)
p_x = predict(k, x)
  lines(x,p_x$mean,col='blue')
  lines(x,p_x$lower95,col='blue')
  lines(x,p_x$upper95,col='blue')
newX <- as.matrix(runif(3))
newy <- f(newX)
  points(newX,newy,col='red')
update(k,newy,newX)
x = seq(0,1,,101)
p2_x = predict(k, x)
  lines(x,p2_x$mean,col='red')
  lines(x,p2_x$lower95,col='red')
  lines(x,p2_x$upper95,col='red')
```

update.Kriging *Update Kriging model with new points*

Description

Update Kriging model with new points

Usage

```
## S3 method for class 'Kriging'
update(object, newy, newX, normalize = FALSE, ...)
```

Arguments

`object`	S3 Kriging object
`newy`	new points in model output space
`newX`	new points in model input space
`normalize`	Normalize X and y in [0,1] (FALSE by default)
`...`	Ignored

Author(s)

Yann Richet (yann.richet@irsn.fr)

Examples

```
f = function(x) 1-1/2*(sin(12*x)/(1+x)+2*cos(7*x)*x^5+0.7)
  plot(f)
set.seed(123)
X <- as.matrix(runif(5))
y <- f(X)
  points(X,y,col='blue')
r <- Kriging(y, X, "gauss")
x = seq(0,1,,101)
p_x = predict(r, x)
  lines(x,p_x$mean,col='blue')
  lines(x,p_x$mean-2*p_x$stdev,col='blue')
  lines(x,p_x$mean+2*p_x$stdev,col='blue')
newX <- as.matrix(runif(3))
newy <- f(newX)
  points(newX,newy,col='red')
update(r,newy,newX)
x = seq(0,1,,101)
p2_x = predict(r, x)
  lines(x,p2_x$mean,col='red')
  lines(x,p2_x$mean-2*p2_x$stdev,col='red')
  lines(x,p2_x$mean+2*p2_x$stdev,col='red')
```