Package: quantkriging (via r-universe)

September 17, 2024

Type Package

Title Quantile Kriging for Stochastic Simulations with Replication

Version 0.1.0

Author Kevin R. Quinlan [aut, cre], Jim R. Leek [aut], Lawrence Livermore National Security [cph]

Maintainer Kevin R. Quinlan <quinlan5@llnl.gov>

Description A re-implementation of quantile kriging. Quantile kriging was described by Plumlee and Tuo (2014) <doi:10.1080/00401706.2013.860919>. With computational savings when dealing with replication from the recent paper by Binois, Gramacy, and Ludovski (2018) <doi:10.1080/10618600.2018.1458625> it is now possible to apply quantile kriging to a wider class of problems. In addition to fitting the model, other useful tools are provided such as the ability to automatically perform leave-one-out cross validation.

Depends R (>= 3.6.0)

Imports hetGP (>= 1.1.1), Matrix (>= 1.2.17), ggplot2 (>= 3.2.1), reshape2 (>= 1.4.3), stats

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests testthat (>= 2.1.0)

NeedsCompilation no

Date/Publication 2020-03-06 13:20:02 UTC

Repository https://kevinquinlan.r-universe.dev

RemoteUrl https://github.com/cran/quantkriging

RemoteRef HEAD

RemoteSha c959e5dd1e0019bbff2ae064db1fac6f1288bbb5

Contents

LOOquants 2
newQuants 3
new_QKResults 3
predict.QKResults 4
quantKrig 5
quantkriging 7
QuantPlot 7

Index **9**

`LOOquants` *Revaluate Quantiles*

Description

Generates Leave-One-Out predictions for each location and quantile.

Usage

```
LOOquants(QKResults)
```

Arguments

`QKResults`	Output from the quantKrig function.

Details

Returns the estimated quantiles and a plot of the leave-one-out predictions against the observed values.

Value

Leave-one-out predictions at the input locations

Examples

```
X <- seq(0,1,length.out = 20)
Y <- cos(5*X) + cos(X)
Xstar <- rep(X,each = 100)
Ystar <- rep(Y,each = 100)
e <- rchisq(length(Ystar),5)/5 - 1
Ystar <- Ystar + e
lb <- c(0.0001,0.0001)
ub <- c(10,10)
Qout <- quantKrig(Xstar,Ystar, seq(0.05,0.95, length.out = 7), lower = lb, upper = ub)

LOOquants(Qout)
```

newQuants *Revaluate Quantiles*

Description

Generates new quantiles from a quantile object

Usage

```
newQuants(QKResults, quantv)
```

Arguments

QKResults	Output from the quantKrig function.
quantv	Vector of quantile values alpha between 0 and 1,

Value

The same quantile object with new estimated quantiles.

Examples

```
X <- seq(0,1,length.out = 20)
Y <- cos(5*X) + cos(X)
Xstar <- rep(X,each = 100)
Ystar <- rep(Y,each = 100)
e <- rchisq(length(Ystar),5)/5 - 1
Ystar <- Ystar + e
lb <- c(0.0001,0.0001)
ub <- c(10,10)
Qout <- quantKrig(Xstar,Ystar, seq(0.05,0.95, length.out = 7), lower = lb, upper = ub)

Qout2 <- newQuants(Qout, c(0.025, 0.5, 0.975))
QuantPlot(Qout2)
```

new_QKResults *QKResult Constructor*

Description

Create Quantile Kriging Results class from list

Usage

```
new_QKResults(qkList)
```

Arguments

qkList	list(quants, yquants, g, l, ll <- -optparb$value, beta0, nu, xstar, ystar, Ki, quantv, mult, ylisto, type)

Value

New class QKResults

---

predict.QKResults *Revaluate Quantiles*

Description

Quantile Predictions using Quantile Kriging model (class QKResults)

Usage

```
## S3 method for class 'QKResults'
predict(object, xnew, quantnew = NULL, ...)
```

Arguments

object	Output from the quantKrig function.
xnew	Locations for prediction
quantnew	Quantiles for prediction, default is to keep the same as the quantile object
...	Ignore. No other arguments for this method

Value

Quantile predictions at the specified input locations

Author(s)

Kevin Quinlan quinlan5@llnl.gov

Examples

```
X <- seq(0,1,length.out = 20)
Y <- cos(5*X) + cos(X)
Xstar <- rep(X,each = 100)
Ystar <- rep(Y,each = 100)
e <- rchisq(length(Ystar),5)/5 - 1
Ystar <- Ystar + e
lb <- c(0.0001,0.0001)
ub <- c(10,10)
Qout <- quantKrig(Xstar,Ystar, seq(0.05,0.95, length.out = 7), lower = lb, upper = ub)
```

```
predict(Qout, xnew = c(0.4, 0.5, 0.6))

quantpreds <- predict(Qout, xnew = seq(0,1,length.out = 100), quantnew = seq(0.01,0.99,by = 0.01))
matplot(seq(0,1,length.out = 100), quantpreds, type = 'l')
```

---

quantKrig *Quantile Kriging*

Description

Implements Quantile Kriging from Plumlee and Tuo (2014).

Usage

```
quantKrig(x, y, quantv, lower, upper, method = "loo",
  type = "Gaussian", rs = TRUE, nm = TRUE, known = NULL,
  optstart = NULL, control = list())
```

Arguments

x	Inputs
y	Univariate Response
quantv	Vector of Quantile values to estimate (ex: c(0.025, 0.975))
lower	Lower bound of hyperparameters, if isotropic set lengthscale then nugget, if anisotropic set k lengthscales and then nugget
upper	Upper bound of hyperparameters, if isotropic set lengthscale then nugget, if anisotropic set k lengthscales and then nugget
method	Either maximum likelihood ('mle') or leave-one-out cross validation ('loo') optimization of hyperparameters
type	Covariance type, either 'Gaussian', 'Matern3_2', or 'Matern5_2'
rs	If TRUE, rescales inputs to [0,1]
nm	If TRUE, normalizes output to mean 0, variance 1
known	Fixes all hyperparamters to a known value
optstart	Sets the starting value for the optimization
control	Control from optim function

Details

Fits quantile kriging using a double exponential or Matern covariance function. This emulator is for a stochastic simulation and models the distribution of the results (through the quantiles), not just the mean. The hyperparameters can be trained using maximum likelihood estimation or leave-one-out cross validation as recommended in Plumlee and Tuo (2014). The GP is trained using the Woodbury formula to improve computation speed with replication as shown in Binois et al. (2018). To get

meaningful results, there should be sufficient replication at each input. The quantiles at a location $x0$ are found using:

$$\mu(x0) + kn(x0)Kn^{-1}(y(i) - \mu(x)$$

) where Kn is the kernel of the design matrix (with nugget effect), $y(i)$ the ordered sample closest to that quantile at each input, and $\mu(x)$ the mean at each input.

Value

quants The estimated quantile values in matrix form

yquants The actual quantile values from the data in matrix form

g The scaling parameter for the kernel

l The lengthscale parameter(s)

ll The log likelihood

beta0 Estimated linear trend

nu Estimator of the variance

xstar Matrix of unique input values

ystar Average value at each unique input value

Ki Inverted covariance matrix

quantv Vector of alpha values between 0 and 1 for estimated quantiles, it is recommended that only a small number of quantiles are used for fitting and more quantiles can be found later using newQuants

mult Number of replicates at each input

References

- Matthew Plumlee & Rui Tuo (2014) Building Accurate Emulators for Stochastic Simulations via Quantile Kriging, Technometrics, 56:4, 466-473, DOI: 10.1080/00401706.2013.860919
- Mickael Binois, Robert B. Gramacy & Mike Ludkovski (2018) Practical Heteroscedastic Gaussian Process Modeling for Large Simulation Experiments, Journal of Computational and Graphical Statistics, 27:4, 808-821, DOI: 10.1080/10618600.2018.1458625

Examples

```
# Simple example
X <- seq(0,1,length.out = 20)
Y <- cos(5*X) + cos(X)
Xstar <- rep(X,each = 100)
Ystar <- rep(Y,each = 100)
Ystar <- rnorm(length(Ystar),Ystar,1)
lb <- c(0.0001,0.0001)
ub <- c(10,10)
Qout <- quantKrig(Xstar,Ystar, quantv = seq(0.05,0.95, length.out = 7), lower = lb, upper = ub)
QuantPlot(Qout, Xstar, Ystar)

#fit for non-normal errors
```

```
Ystar <- rep(Y,each = 100)
e <- rchisq(length(Ystar),5)/5 - 1
Ystar <- Ystar + e
Qout <- quantKrig(Xstar,Ystar, quantv = seq(0.05,0.95, length.out = 7), lower = lb, upper = ub)
QuantPlot(Qout, Xstar, Ystar)
```

---

quantkriging *quantkriging*

Description

Quantile Kriging is a method to model the uncertainty of a stochastic simulation by modelling both the overall simulation response and the output distribution at each sample point. The output distribution is characterized by dividing it into quantiles, where the division of each quantile is determined by kriging.

Details

This library is our re-implementation of Quantile Kriging as described by Matthew Plumlee & Rui Tuo in their 2014 paper "Building Accurate Emulators for Stochastic Simulations via Quantile Kriging." With computational savings when dealing with replication from the recent paper "Practical heteroskedastic Gaussian process modeling for large simulation experiments " by Binois, M., Gramacy, R., and Ludovski, M. it is now possible to apply Quantile Kriging to a wider class of problems. In addition to fitting the model, other useful tools are provided such as the ability to automatically perform leave-one-out cross validation.

quantkriging functions

- quantKrig : Implements Quantile Kriging
- quantPlot : Plots the Quantile output from quantKrig if there is only one input.
- newQuants : Generates new quantiles from a quantile object
- LOOquants : Generates Leave-One-Out predictions for each location and quantile.

QuantPlot *Plot Univariate Quantile Data*

Description

Plots the Quantile output from quantKrig if there is only one input.

Usage

```
QuantPlot(QKResults, X1 = NULL, Y1 = NULL, main = NULL,
  xlab = NULL, ylab = NULL, colors = NULL)
```

Arguments

QKResults	Output from the quantKrig function.
X1	X values if ploting the original data in the background
Y1	Y values if ploting the original data in the background
main	Plot Title defaults to Fitted Quantiles
xlab	Label for x-axis defaults to X
ylab	Label for y-axis defaults to Y
colors	Customize colors associated with the quantiles

Value

A ggplot object

Examples

```
X <- seq(0,1,length.out = 20)
Y <- cos(5*X) + cos(X)
Xstar <- rep(X,each = 100)
Ystar <- rep(Y,each = 100)
Ystar <- rnorm(length(Ystar),Ystar,1)
Ystar <- (Ystar - mean(Ystar)) / sd(Ystar)
Xstar <- (Xstar - min(Xstar)/ max(Xstar) - min(Xstar))
lb <- c(0.0001,0.0001)
ub <- c(10,10)
Qout <- quantKrig(Xstar,Ystar, seq(0.05,0.95, length.out = 7), lower = lb, upper = ub)
QuantPlot(Qout, Xstar, Ystar)
```

Index

LOOquants, 2

new_QKResults, 3
newQuants, 3

predict.QKResults, 4

quantKrig, 5
quantkriging, 7
quantkriging-package (quantkriging), 7
QuantPlot, 7