Package 'scp'

April 13, 2020

Type Package

Title Spatial Conformal Prediction

Version 0.1.0

Author Maggie Mao <hmao@samsi.info>

Maintainer Maggie Mao <hmao@samsi.info>

Description Provide spatial prediction intervals
using Global Spatial Conformal Prediction (GSCP)
and Local Spatial Conformal Prediction (LSCP)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

URL https://github.com/mhuiying/scp

BugReports https://github.com/mhuiying/scp/issues

Suggests knitr,
rmarkdown,
covr,
testthat

VignetteBuilder knitr

Imports geoR

R topics documented:

compute_UVW *Internal function: compute U, V, W in Appendix A*

Description

Internal function: compute U, V, W in Appendix A

Usage

```
compute_UVW(Q, Y)
```

Arguments

Q	inverse of the covariance matrix
Y	a vector with n data values.

Value

a list of U, V, W values

conformal_pred *Spatial Conformal Prediction Intervals*

Description

This function provides the conformal prediction interval for spatial location `s0` given obserations `s,Y`.

Usage

```
conformal_pred(s0, s, Y, theta, eta = Inf, m = NULL, alpha = 0.05)
```

Arguments

s0	prediction location
s	an $n \times d$ matrix or data-frame with d coordinates of the n data locations.
Y	a vector with n data values.
theta	spatial covariance parameters as in `mat_cov`
eta	numerical value of the kernel bandwidth for the weight schema in conformal prediction. Defauls to Inf meaning equal weight on surrounding m points.
m	an postive integer representing the number of nearest locations used for prediction. Depends on eta.
alpha	significance level. Defaults to 0.05.

Value

A vector of lower and upper bounds of the conformal prediction interval.

Examples

```
N = 40; n = N^2
S = seq(0,1,length=N)
s <- expand.grid(S,S)
d <- as.matrix(dist(s))

theta        = c(0,3,0.1,0.7)
names(theta) = c("Nugget","PartialSill","Range","Smoothness")
C <- mat_cov(d,theta)
X <- t(chol(C))%*%rnorm(n)
Y <- X^3 + rnorm(n)

# Estimate spatial covariance parameters
bins     <- seq(0.01,0.2,0.01)
thetaHat <- get_theta(s,Y,dists=bins,plot_fitted=FALSE)
Q        <- solve(mat_cov(d,thetaHat))

s0       <- c(0.5, 0.5)
conformal_pred(s0,s,Y,thetaHat,m=100,eta=0.1)
```

`fast_scp` *Fast Spatial Conformal Prediction Intervals*

Description

Internal function provides the conformal prediction interval for spatial location `s0` given partial obserations `s1,...,sm,Y1,..,Ym` when the square of the standard kriging residuals are used as the non-conformality measures, and weights are provided.

Usage

```
fast_scp(s0, s, Y, thetaHat, alpha = 0.05)
```

Arguments

`s0`	prediction location
`s`	an $n \times d$ matrix or data-frame with d coordinates of the n data locations.
`Y`	a vector with n data values.
`thetaHat`	estimated Matern covariance parameters as in `mat_cov`
`alpha`	significance level. Defaults to 0.05.
`w`	weights for the non-conformity measures.

Value

A vector of lower and upper bounds of the conformal prediction interval.

get_theta *Estimate theta via variogram fitting*

Description

Estimate theta via variogram fitting

Usage

```
get_theta(s, Y, dists = NULL, plot_fitted = FALSE)
```

Arguments

s	spatial location
Y	data
dists	breakpoints for bins
plot_fitted	will plot a emperical variogram if specified as TRUE

Value

Matern covariance parameters

krige_pred *Kriging Prediction Function*

Description

Kriging Prediction Function

Usage

```
krige_pred(
  s0,
  s,
  Y,
  alpha = 0.05,
  thetaHat = NULL,
  interval = FALSE,
  return_sd = FALSE
)
```

Arguments

s0	prediction location
s	an $n \times d$ matrix or data-frame with d coordinates of the n data locations.
Y	a vector with n data values.
alpha	significance level. Defaults to 0.05.
thetaHat	spatial covariance parameters as in mat_cov. If not given, emperical variogram is used to estimate thetaHat.
interval	logical; if TRUE, Krige_pred function returns prediction interval; if FALSE, Krige_pred function returns point prediction interval. Defaults to FALSE.
return_sd	logical; if TRUE, Krige_pred function returns standard deviation along with the point prediction. Defaults to FALSE.

Value

a value of point prediction if interval is FALSE or a vector of lower and upper bounds of the Kriging prediction intervalif $interval$ is TRUE.

Examples

```
N = 21; n = N^2
S = seq(0,1,length=N)
s = expand.grid(S,S)
d = as.matrix(dist(s))

theta        = c(0,3,0.1,0.7)
names(theta) = c("Nugget","PartialSill","Range","Smoothness")
C = mat_cov(d,theta)
X = t(chol(C))%*%rnorm(n)
Y = X^3 + rnorm(n)

s0 = c(0.5, 0.5)
krige_pred(s0,s,Y)
krige_pred(s0,s,Y,interval=TRUE)
```

---

mat_cov *Matern covariance function*

Description

Matern covariance function

Usage

```
mat_cov(d, theta)
```

Arguments

d	a numeric distance, a vector of distances, or a distance matrix
theta	Matern covariance parameters

Value

a numeric covariance, a vector of covariances, or a covariance matrix in the same size of d

---

scp *Spatial Conformal Prediction (SCP) At a Single Input Location*

Description

This function provides the conformal prediction interval for spatial location `s0` given obserations `s,Y`.

Usage

```
scp(
  s0,
  s,
  Y,
  global = FALSE,
  eta = Inf,
  m = NULL,
  pred_fun = krige_pred,
  alpha = 0.05,
  dfun = "std_residual2",
  precision = 0.01
)
```

Arguments

s0	prediction location
s	an $n \times d$ matrix or data-frame with d coordinates of the n data locations.
Y	a vector with n data values.
global	logical; if TRUE , `scp` function returns the result of global spatial conformal prediction `gscp`; if FALSE, `scp` function returns the result of local spatial conformal prediction `lscp`.
eta	numerical value of the kernel bandwidth for the weight schema in conformal prediction. Defauls to Inf meaning equal weight on surrounding m points.
m	an postive integer representing the number of nearest locations used for prediction. Depends on eta.
pred_fun	spatial point prediction function
alpha	significance level. Defaults to 0.05.
dfun	non-conformity measure
precision	Defaults to 0.01.

Value

A vector of lower and upper bounds of the conformal prediction interval.

Examples

```
N = 41; n = N^2
S = seq(0,1,length=N)
s = expand.grid(S,S)
d = as.matrix(dist(s))

theta        = c(0,3,0.1,0.7)
names(theta) = c("Nugget","PartialSill","Range","Smoothness")
C = mat_cov(d,theta)
X = t(chol(C))%*%rnorm(n)
Y = X^3 + rnorm(n)

s0 = c(0.5, 0.5)
idx = which(s[,1]==s0[1] & s[,2]==s0[2])
pred_fun = function(s0,s,Y,alpha=0.05) return(mean(Y))
PI = scp(s0,s[-idx,],Y[-idx],pred_fun=pred_fun, dfun="abs_residual",precision=0.1)
cat(paste("True value: ", Y[idx], "\n"))
cat(paste("Prediction Interval: [ ", PI[1], ",", PI[2], "]"))
```

Index