

Package ‘scp’

April 26, 2020

Type Package

Title Spatial Conformal Prediction

Version 0.1.0

Author Huiying Mao <hmao@samsi.info>, Brian Reich <bjreich@ncsu.edu>, Ryan Martin <rgmarti3@ncsu.edu>

Maintainer Huiying Mao <hmao@samsi.info>

Description Provide robust, valid, and model-free spatial prediction intervals using Spatial Conformal Prediction (SCP) algorithms

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:

krige_pred	2
plausibility	3
plausibility_contour	4
scp	6

Index	8
-------	---

krige_pred	<i>Kriging prediction function</i>
------------	------------------------------------

Description

This function provides an example for `pred_fun` in [scp](#), [plausibility](#), and [plausibility_contour](#), which provides a point prediction for location `s0` (and corresponding standard error), given observations `s` and `Y`.

Usage

```
krige_pred(s0, s, Y, return_sd = FALSE)
```

Arguments

<code>s0</code>	prediction location, a numeric vector with <code>length = 2</code> .
<code>s</code>	an $n \times 2$ matrix or <code>data.frame</code> with two coordinates of n locations.
<code>Y</code>	a vector with n values corresponding to $Y(s)$.
<code>return_sd</code>	logical. if TRUE, <code>Krige_pred</code> function returns the standard error of $Y(s0)$ along with the point prediction; if FALSE, <code>Krige_pred</code> function only returns the point prediction. Defaults to FALSE.

Value

The output is a value of point prediction for $Y(s0)$ if `return_sd` is FALSE or a list with the following elements if `return_sd` is TRUE.

<code>yhat</code>	point prediction for $Y(s0)$
<code>sd</code>	standard error for $Y(s0)$

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,return_sd=TRUE)
```

plausibility	<i>calculate plausibility for Y_0</i>
--------------	----------------------------------------------------

Description

This function provides the plausibility of $Y(s_0)$ being Y_0 , given observations s and Y , using spatial conformal prediction algorithms.

Usage

```
plausibility(
  Y0,
  s0,
  s,
  Y,
  global = TRUE,
  eta = Inf,
  m = NULL,
  pred_fun = krige_pred,
  dfun = c("residual2", "abs_residual", "std_residual2", "std_abs_residual")
)
```

Arguments

Y_0	a scalar or a vector
s_0	prediction location, a numeric vector with <code>length = 2</code> .
s	an $n \times 2$ matrix or data.frame with two coordinates of n locations.
Y	a vector with n values corresponding to $Y(s)$.
global	logical; if TRUE , <code>scp</code> function returns the result of global spatial conformal prediction (GSCP); if FALSE, <code>scp</code> function returns the result of local spatial conformal prediction (LSCP) and users need to specify <code>eta < Inf</code> or $m \leq n$. Defaults to TRUE.
eta	kernel bandwidth for weight schema, a positive scalar with smaller value meaning more localized procedure. Defaults to Inf, which puts equal weight on surrounding m points.
m	an positive integer representing the number of nearest locations to use for prediction. Default to NULL. If <code>global = TRUE</code> , $m = n$; if <code>global = FALSE</code> and m is not specified, m would be determined by <code>eta</code> .
pred_fun	spatial prediction function with inputs being <code>s0,s,Y</code> and ouputs being predicted $Y(s_0)$ (and its standard error). Defaults to krige_pred .
dfun	non-conformity measure with four options. In which, "residual2" (default) represents squared residual, "std_residual2" represents standardized squared residual, "abs_residual" represents absolute residual, and "std_abs_residual" represents standardized absolute residual.

Value

The output is a scalar or a vector with plausibility values for Y_0 . The numbers are between 0 and 1.

Author(s)

Huiying Mao, <hmao@samsi.info>, Brian Reich <bjreich@ncsu.edu>

References

to be entered

See Also

[plausibility_contour](#)

Examples

```
## To predict  $Y(s_0)$ , where  $s_0 = c(0.5, 0.5)$ , using sample data
## What's the plausibility if  $Y(s_0) = 0$ ?  $Y(s_0) = 1.5$ ?

?sample_data
s0 = c(0.5, 0.5)
s = sample_data$s
Y = sample_data$Y

# plausibility for  $Y(s_0) = 0$ : 0.8744795
plausibility(Y0=0, s0=s0, s=s, Y=Y)

# plausibility for  $Y(s_0) = 1.5$ : 0.4669839
plausibility(Y0=1.5, s0=s0, s=s, Y=Y)

# plausibility for a sequence of  $Y_0$ 's
plausibility(Y0=seq(0, 1, 0.1), s0=s0, s=s, Y=Y)
```

plausibility_contour *generate plausibility contour*

Description

This function provides the plausibility contour for $Y(s_0)$, given observations s and Y , using spatial conformal prediction algorithms.

Usage

```
plausibility_contour(
  s0,
  s,
  Y,
  global = TRUE,
  eta = Inf,
  m = NULL,
  pred_fun = krige_pred,
  dfun = c("residual2", "abs_residual", "std_residual2", "std_abs_residual"),
  precision = NULL
)
```

Arguments

<code>s0</code>	prediction location, a numeric vector with <code>length = 2</code> .
<code>s</code>	an $n \times 2$ matrix or data.frame with two coordinates of n locations.
<code>Y</code>	a vector with n values corresponding to $Y(s)$.
<code>global</code>	logical; if <code>TRUE</code> , <code>scp</code> function returns the result of global spatial conformal prediction (GSCP); if <code>FALSE</code> , <code>scp</code> function returns the result of local spatial conformal prediction (LSCP) and users need to specify <code>eta < Inf</code> or <code>m ≤ n</code> . Defaults to <code>TRUE</code> .
<code>eta</code>	kernel bandwidth for weight schema, a positive scalar with smaller value meaning more localized procedure. Defaults to <code>Inf</code> , which puts equal weight on surrounding <code>m</code> points.
<code>m</code>	an positive integer representing the number of nearest locations to use for prediction. Default to <code>NULL</code> . If <code>global = TRUE</code> , <code>m = n</code> ; if <code>global = FALSE</code> and <code>m</code> is not specified, <code>m</code> would be determined by <code>eta</code> .
<code>pred_fun</code>	spatial prediction function with inputs being <code>s0, s, Y</code> and outputs being predicted $Y(s0)$ (and its standard error). Defaults to krige_pred .
<code>dfun</code>	non-conformity measure with four options. In which, <code>"residual2"</code> (default) represents squared residual, <code>"std_residual2"</code> represents standardized squared residual, <code>"abs_residual"</code> represents absolute residual, and <code>"std_abs_residual"</code> represents standardized absolute residual.
<code>precision</code>	a positive scalar represents how dense $Y(s)$ candidates (<code>Y_cand</code>) are. Defaults to <code>NULL</code> .

Value

The output is a data.frame of `Y_cand` and corresponding plausibility values

Author(s)

Huiying Mao, <hmao@samsi.info>, Brian Reich <bjreich@ncsu.edu>

References

to be entered

See Also[plausibility](#)

scp

*Spatial conformal prediction at a single input location***Description**

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

Usage

```
scp(
  s0,
  s,
  Y,
  global = TRUE,
  eta = Inf,
  m = NULL,
  pred_fun = krige_pred,
  dfun = c("residual2", "abs_residual", "std_residual2", "std_abs_residual"),
  precision = NULL,
  alpha = 0.05
)
```

Arguments

<code>s0</code>	prediction location, a numeric vector with <code>length = 2</code> .
<code>s</code>	an $n \times 2$ matrix or <code>data.frame</code> with two coordinates of n locations.
<code>Y</code>	a vector with n values corresponding to $Y(s)$.
<code>global</code>	logical; if <code>TRUE</code> , <code>scp</code> function returns the result of global spatial conformal prediction (GSCP); if <code>FALSE</code> , <code>scp</code> function returns the result of local spatial conformal prediction (LSCP) and users need to specify <code>eta</code> . Defaults to <code>TRUE</code> .
<code>eta</code>	kernel bandwidth for weight schema, a positive scalar with smaller value meaning more localized procedure. Defaults to <code>Inf</code> , which puts equal weight on surrounding m points.
<code>m</code>	an positive integer representing the number of nearest locations to use for prediction. Default depends on <code>eta</code> .
<code>pred_fun</code>	spatial prediction function with inputs being <code>s0, s, Y</code> and outputs being predicted $Y(s0)$ (and its standard error). Defaults to krige_pred representing Kriging prediction.

<code>dfun</code>	non-conformity measure with four options. In which, "residual2" (default) represents squared residual, "std_residual2" represents standardized squared residual, "abs_residual" represents absolute residual, and "std_abs_residual" represents standardized absolute residual.
<code>precision</code>	a positive scalar represents how dense the candidates for $Y(s)$ are. Defaults to NULL.
<code>alpha</code>	significance level. Defaults to 0.05.

Value

The output is a vector of lower and upper bounds of the conformal prediction interval.

Author(s)

Huiying Mao, <hmao@samsi.info>, Brian Reich <bjreich@ncsu.edu>

References

to be entered

See Also

[plausibility](#), [plausibility.contour](#)

Examples

```
## generate prediction interval for s0 = c(0.5,0.5) using sample data

?sample_data
s0 = c(0.5,0.5)
s = sample_data$s
Y = sample_data$Y

# default prediction interval, output -12.81192 12.14449
scp(s0=s0,s=s,Y=Y)

# user define eta=0.1, where LSCP is considered. output -5.891513 5.072357
scp(s0=s0,s=s,Y=Y,eta=0.1)

# user define non-conformity measure. output -16.27205 15.12093
scp(s0=s0,s=s,Y=Y,dfun="abs_residual")

# user define prediction function
fun = function(s0,s,Y) return(mean(Y)). output -23.65864 26.20081
scp(s0=s0,s=s,Y=Y,pred_fun=fun)
```

Index

krige_pred, [2](#), [3](#), [5](#), [6](#)

plausibility, [2](#), [3](#), [6](#), [7](#)

plausibility_contour, [2](#), [4](#), [4](#), [7](#)

scp, [2](#), [6](#)