Package ‘STMedianPolish’

March 8, 2017

Type Package
Title Spatio-Temporal Median Polish
Version 0.2
Date 2017-03-07
Author William Martínez [aut, cre], Melo Carlos [aut], Melo Oscar [aut].
Maintainer William Martínez <willimarti2008@gmail.com>
Depends R (>= 2.15)
Imports maptools, reshape2, sp, spacetime, zoo, nabor, gstat
URL https://github.com/WilliamAMartinez/STMedianPolish
Description Analyses spatio-temporal data, decomposing data in n-dimensional arrays and using the median polish technique.
License GPL (>= 2)
LazyData true
Encoding UTF-8
RoxygenNote 6.0.1
NeedsCompilation no
Repository CRAN
Date/Publication 2017-03-08 08:22:14

R topics documented:

ConstructMPst ................................................. 2
DemMeta ..................................................... 3
HZRMeta ...................................................... 4
krige0STlocalMP .............................................. 4
MedianPolishM .................................................. 6
MedianPolishM.ConstructMPst ............................... 7
MedianPolishM.default ..................................... 8
Metadb ....................................................... 9
Mpplot ....................................................... 10
ConstructMPst

ConstructMPst

Description

Create an spatio-temporal object with regular data, in order to employ median polish technique.

Usage

ConstructMPst(valuest, time, pts, Delta)

Arguments

valuest: data.frame in which different columns refer to different locations, and each row reflects a particular observation time.

time: indicate the time of valuest, the intervals of time must be regular.

pts: data.frame that hold three dimensions spatial coordinates x, y and z. Therefore, the coordinates should be projected.

Delta: vector with number of divisions of each spatial direction. c(Delta x, Delta y, Delta z).

Details

This function configures an irregular distribution of spatio-temporal data in four-ways. Therefore, the new data corresponds to the average of values and coordinates of every spatio-temporal cell.

Value

An object of class ConstructMPst with the following list of components:

results: average value on the stations set into unity spatio-temporal defined for delta.

Value: array with the results organized by cells, the size of the cells is defined in Delta.

valuest: valuest.

pts: pts.

time: time.

Delta: Delta.
References


Examples

```r
## Not run:
library(zoo)
data(Metadb)
#records of monthly precipitation from january 2007 to january 2010
Metadb<-Metadb[,c(1:4,89:125)]
x<-matrix(0,1,37)
for(i in 1:37){
x[,i] <- 2007 + (seq(0, 36)/12)[i]
}
x<-as.Date (as.yearmon(x), frac = 1)
time = as.POSIXct(x, tz = "GMT")
MPST<-ConstructMPst(Metadb[,]c(1:4)],time,pts=Metadb[,]2:4],Delta=c(7,6,5))
## End(Not run)
```

DemMeta

*Digital Elevation Model Resolution 90 meters.*

Description

Digital elevation model with resolution 250 meters of Hydrogeological zone west of Meta river.
Spatial reference system: Datum Magna Sirgas Origen Bogota.

Usage

```r
data(DemMeta)
```

Format

Formal class: 'SpatialGridDataFrame' [package "sp"]

Source


Examples

```r
library(sp)
data(DemMeta)
Gridxy<- spsample(DemMeta, cellsize=2000, n=300,"regular")
plot(Gridxy)
```
**HZRMeta**

*Hydrogeological zone west of Meta river.*

---

**Description**

Map of hydrogeological zone west of Meta river. Spatial reference system: Datum Magna Sirgas Origen Bogota.

**Usage**

```r
data(HZRMeta)
```

**Format**

The format is: Formal class 'SpatialPolygonsDataFrame' [package "sp"]

**Source**

[http://www.arcgis.com/home/item.html?id=103b63d9c9f448abed63f22b728b1a02](http://www.arcgis.com/home/item.html?id=103b63d9c9f448abed63f22b728b1a02)

**Examples**

```r
library(sp)
data(HZRMeta)
Gridxy<- spsample(HZRMeta, cellsize=2000, n=300,"regular")
plot(Gridxy)
```

---

**krige0STlocalMP**

*Ordinary local Spatio-temporal Kriging*

---

**Description**

Function for ordinary local spatio-temporal kriging

**Usage**

```r
krige0STlocalMP(data,newdata,p,model,k,stiAni)
```

**Arguments**

- `data` object of class 'STFDF' [package "spacetime"]; It must contain the spatio–temporal coordinates and values.
- `newdata` object of class 'STF' [package "spacetime"]; It should contain the prediction location in space and time.
- `p` parameters of the spatio-temporal covariance model. The first parameter must be nugget value.
krige0STlocalMP

model spatio–temporal covariance model.
k defines the number of the input spatio–temporal points that will be used to interpolate one new value.
stAni Constant of the spatio–temporal anisotropy, assuming a metric spatio–temporal space.

Value
Table that contains the prediction and the prediction variance.

References

Examples

library(spacetime)
library(sp)
library(gstat)
library(zoo)
library(maptools)
data(Metadata)
#records of the precipitation monthly from january 2007 to january 2010
Metadata<-Metadata[,c(1:4,89:125)]
x<-matrix(0,1,37)
for(i in 1:37){
  x[,i] <- 2007 + (seq(0, 36)/12)[i]
}
x<-as.Date(as.yearmon(x), frac = 1)
time = as.POSIXct(x, tz = "GMT")

MPST<-ConstructMPst(sqrt(0.5*Metadata[,c(1:4)]),time,pts=Metadata[,2:4],Delta=c(7,6,5))
residual<-removeMRMst(MPST,eps=0.01, maxiter=2)

rain_loc<-Metadata[,c("Station","East","North","Height")]
coordinates(rain_loc) = ~East+North+Height
proj4string(rain_loc) = CRS(proj4string(DemMeta))

rain_residual = stConstruct(data.frame(Res=residual[,7]), space = list(values = 1),
                          time, SpatialObj = rain_loc,interval=TRUE)

#NewData
data(HZRMeta)
polygon1 = polygons(HZRMeta)
Gridxy<- spsample(polygon1, cellsize=10000, n=1000,"regular")
Gridxyz<-data.frame(Gridxy,over(Gridxy,DemMeta))
colnames(Gridxyz)<-c("East", "North","height")
An additive model for multidimensional array is fitted, using Tukey’s median polish procedure.

Usage

MedianPolishM(data, ...)

Arguments

data object of class array, table or matrix (see details).

... default arguments, see MedianPolishM.default

Details

The function MedianPolishM is generic. See the documentation for MedianPolishM.default for further details.

Value

An object of class medpolish with the following named components in a list:

residuals the residuals.
overall the fitted constant term.
effects the fitted every dimensions effects to array multidimensional.
iter number of iterations used in the range maxiter.
References


Description

An additive model for multidimensional array is fitted, using Tukey’s median polish procedure.

Usage

```r
## S3 method for class 'ConstructMPst'
MedianPolishM(data, eps, maxiter, na.rm, ...)
```

Arguments

data class `ConstructMPst`.
eps real number greater than 0, default 0.01. A tolerance for convergence: see Details
maxiter the maximum number of iterations. Default 10.
na.rm logical. If the data contains NA's. Default TRUE.
... ignored.

Details

The model fitted is an additive, $\mu + \alpha_a + \beta_b + \xi_c + \tau_t$, where $\mu$ is an overall mean, $\alpha_a$ is the $a$-th row effect, $\beta_b$ is the effect $b$-th column effect, $\xi_c$ is the $c$-th layer effect, $\tau_t$ is the $t$-th time effect. The algorithm works by alternately removing medians of every spatio - temporal dimensions, and continues until the proportional reduction in the sum of absolute residuals is less than eps or until there have been maxiter iterations. If na.rm is FALSE, the presence of any NA value in x will cause an error, otherwise NA values are ignored. MedianPolishM returns an object of class MedianPolishM (see below). There is a plotting method for this class, `plot.MedianPolishM`.

Value

An object of class medpolish with the following named components in a list:

- `residuals` the residuals.
- `overall` the fitted constant term.
- `effects` the fitted every space - time effects.
- `iter` number of iterations used in the range maxiter.
References


---

**MedianPolishM.default**  
*Median polish multidimensional.*

**Description**

An additive model for multidimensional array is fitted, using Tukey's median polish procedure.

**Usage**

```r
## Default S3 method:
MedianPolishM(data, eps = 0.01, maxiter = 10L,
   na.rm = TRUE, ...)
```

**Arguments**

- `data`  
  object of class array, table or matrix (see details).
- `eps`  
  real number greater than 0, default 0.01. A tolerance for convergence: see Details
- `maxiter`  
  the maximum number of iterations. Default 10.
- `na.rm`  
  logical. If the data contains NA's. Default TRUE.
- `...`  
  ignored.

**Details**

The model fitted is additive $constant + dim_1 + dim_2 + \cdots + dim_n$. The algorithm works by alternately removing medians of $dim_1, \ldots, dim_n$, and continues until the proportional reduction in the sum of absolute residuals is less than `eps` or until there have been `maxiter` iterations. If `na.rm` is FALSE, the presence of any NA value in `x` will cause an error, otherwise NA values are ignored. `MedianPolishM` returns an object of class `MedianPolishM` (see below). There is a plotting method for this class, `plot.MedianPolishM`.

**Value**

An object of class `medpolish` with the following named components in a list:

- `residuals`  
  the residuals.
- `overall`  
  the fitted constant term.
- `effects`  
  the fitted every dimensions effects of array multidimensional.
- `iter`  
  number of iterations used in the range `maxiter`.
References


Examples

A <- MedianPolish(UCBAdmissions, eps=0.1, maxiter=2, na.rm=TRUE)
plot(A)

<table>
<thead>
<tr>
<th>Metadb</th>
<th>Monthly precipitation Meta.</th>
</tr>
</thead>
</table>

Description

Records of 102 pluviometrics station of the 'Instituto de Hidrologia, Meteorologia y Estudios Ambientales de Colombia' (IDEAM), to the west of hidrological zone Meta river. Every station has 121 records of the precipitation monthly from january 2000 to january 2010.

Usage

data(Metadb)

Format

The format is: formal class 'data.frame'

Source

http://www.ideam.gov.co/

Examples

data(Metadb)
str(Metadb)
names(Metadb)
Description

Visualization of the spatial distribution according with three perspectives. Each face has the distribution for trace x, y and z (see ConstructMPst).

Usage

mpplot(MpData)

Arguments

MpData  object of class ConstructMPst.

Value

Graphic of the three perspectives for space data "x", "y", "z", with divisions that contain the number of points in each quadrat.

Examples

library(zoo)
data(Metadata)
# records of monthly precipitation from january 2007 to january 2010
Metadata<-Metadata[,c(1:4,89:125)]
x<-matrix(0,1,37)
for(i in 1:37){
  x[,i] <- 2007 + (seq(0, 36)/12)[i]
}
x<-as.Date (as.yearmon(x), frac = 1)
time = as.POSIXct(x, tz = "GMT")

MPST<-ConstructMPst(Metadata[,c(1:4)],time,pts=Metadata[,2:4],Delta=c(7,6,5))
mpplot(MPST)
**removetrendMPst**

**Usage**

```r
## S3 method for class 'MedianPolishM'
plot(x, ...)
```

**Arguments**

- `x` object of class MedianPolishM.
- `...` ignored.

**Details**

The object of class MedianPolish has a list of the contributions of every effect over data. The graphic shows for each iteration, the behavior of these components. If the median polish is applied to data of class ConstructMPst, this method has a specific graphic for data with space-time variability.

**References**


**Examples**

```r
A <- MedianPolishM(UCBAdmissions, eps=0.1, maxiter=2, na.rm=TRUE)
plot(A)
```

---

### removetrendMPst

**Median polish trend**

**Description**

Direct method to remove the trend of spatio-temporal data through median polish.

**Usage**

```r
removetrendMPst(MPST, eps=0.01, maxiter=10L)
```

**Arguments**

- `MPST` object of class ConstructMPst
- `eps` real number greater than 0, default 0.01. A tolerance for convergence of median polish.
- `maxiter` the maximum number of iterations, default 10.
Details

Following the Berke’s approach (Berke, 2001) to remove spatial trend, Martínez et al. (2017) used a structure of four dimensions to apply median polish technique.

For data \( \{Y(s_{abc}, t), a = 1, ..., U; b = 1, ..., V; c = 1, ..., W; t = 1, ..., n\} \), a spatial and temporal process is given by:

\[
Y(s_{abc}, t) = \mu_y(s_{abc}, t) + \delta_{abct}
\]

where

\[
\mu_y(s_{abc}, t) = \mu + \alpha_a + \beta_b + \xi_c + \tau_t
\]

and \( \delta_{abct} \) is a fluctuation arising from natural variability and from the measurement process. Additionally, \( \mu \) is an overall mean, \( \alpha_a \) is the \( a \)-th row effect, \( \beta_b \) is the effect \( b \)-th column effect, \( \xi_c \) is the \( c \)-th layer effect and \( \tau_t \) is the \( t \)-th time effect.

Value

data.frame with the following fields:

- **ET** indicate the time of a observation.
- **x** spatial coordinate x.
- **y** spatial coordinate y.
- **z** spatial coordinate z.
- **Trend** trend calculated through median polish space-time.
- **Value** observed values.
- **Residual** \( Residual = Value - Trend \).

References


Examples

```r
# Not run:
library(zoo)
data(Metadb)
#records of monthly precipitation from january 2007 to january 2010
Metadb<-Metadb[,c(1:4,89:125)]
x<-matrix(0,1,37)
for(i in 1:37){
  x[,i] <- as.Date(2007 + (seq(0, 36)/12)[i])
}
x<-as.Date(x, frac = 1)
time = as.POSIXct(x, tz = "GMT")

MPST<-ConstructMPst(Metadb[,c(1:4)],time,pts=Metadb[,2:4],Delta=c(7,6,5))
```
splineMPST

Median polish Spline.

Description
The "splineMPST" is designed to represent the variability of spatio-temporal effects on a surface, from robust median polish algorithm and planar interpolation.

Usage
splineMPST(Grid, Ef_t, MPST, eps, maxiter)

Arguments
Grid grid with the coordinates in space "x", "y", "z", where will be viewed trend.
Ef_t temporal scenery to look trend.
MPST object of class ConstructMPst.
eps real number greater than 0, default 0.01. A tolerance for convergence.
maxiter the maximum number of iterations, default 10.

Value
Data frame, where columns show the trend in each spatio-temporal location.

References

Examples
## Not run:
library(zoo)
library(sp)
library(spacetime)
data(Metadb)
# records of monthly precipitation from January 2007 to January 2010
Metadb<-Metadb[,c(1:4,89:125)]
x<-matrix(0,1,37)
for(i in 1:37){
x[,i] <- 2007 + (seq(0, 36)/12)[i]}
residual<-removetrendMPst(MPST,eps=0.01, maxiter=2)
## End(Not run)
{VRes

x<-as.Date(as.yearmon(x), frac = 1)
time = as.POSIXct(x, tz = "GMT")
length(time)
MPST<-ConstructMPSt(Metadb[-,c(1:4)],time,pts=Metadb[,2:4],Delta=c(7,6,5))

MpSTData<-MedianPolish(MPST,eps=0, maxiter=2, na.rm=TRUE)
plot(MpSTData)
data(DemMeta)
xy = SpatialPoints(Metadb[,2:4],CRS(proj4string(DemMeta)))
data(HZRMeta)
polygon1 = polygons(HZRMeta)
Gridxy<- spsample(polygon1, cellsize=3000, n=300,"regular")
Grid<-data.frame(Gridxy,over(Gridxy,DemMeta))
colnames(Grid)<-c("East", "North","height")
TendenciaGrilla<-splineMPST(Grid,EF_t=time[16:21],MPST,eps=0.01, maxiter=2)
IDs = paste("ID",1:nrow(TendenciaGrilla))
mydata = data.frame(values = TendenciaGrilla[,5], ID=IDs)
wind.ST1 = STFDF(SpatialPixels(Gridxy),time[16:21],mydata)
stplot(wind.ST1,col.regions=bpy.colors(40),par.strip.text = list(cex=0.7)
 ,main="Spline median polish: Monthly Precipitation")
## End(Not run)

VRes Precomputed Variogram for residuals of monthly precipitation

Description

Precomputed Variogram for residuals of monthly precipitation Metadb. For this, the 'variogram' [package "gstat"] function is used.

Usage

data(VRes)

Format

The format is: 'StVariogram' 'data.frame'

References


Examples

# Empirical variogram
# VRes = variogram(values ~ 1, rain_residual, cutoff=90000, tlags=0:24, width=2650,
#   assumeRegular=TRUE, na.omit=TRUE)
data(VRes)
plot(VRes)
FitPar_st = function(p, gfn, v, trace = FALSE, ...) {
  mod = gfn(v$spacelag, v$timelag, p, ...)
  resid = v$gamma - mod
  if (trace)
    print(c(p, MSE = mean(resid^2)))
  mean(resid^2)
}
ModSpatial = function(h, p){p[2]*(1-exp(-h/p[3]))}
ModTemporal = function(u,p){p[4]*(1-exp(-u/p[5]))+ p[6]*(1-cos(pi*u/180))+ p[7]*abs(sin(pi*u/180))}
VarioST= function(h,u,p)
  {ModTemporal(u,p)*ModSpatial(h,p)+p[8]*ModTemporal(u,p)*ModSpatial(h,p)}
# Parametros Iniciales
p1<-c(2,14.5,13900,5.9,29,1.55,3.7,-0.07)
pars.st = optim(p1, FitPar_st, method = "BFGS",
   gfn = VarioST, v = VRes, hessian=TRUE)
fit_Vario_ST<-VRes
fit_Vario_ST$gamma<-VarioST(VRes$spacelag, VRes$timelag, pars.st$par)
plot(fit_Vario_ST)
Index

*Topic** datasets
  DemMeta, 3
  HZRMeta, 4
  Metadb, 9
  VRes, 14

ConstructMPst, 2, 7, 10, 11, 13

DemMeta, 3

HZRMeta, 4

krige@STlocalMP, 4

MedianPolishM, 6
MedianPolishM.ConstructMPst, 7
MedianPolishM.default, 6, 8
Metadb, 9, 14
Mpplot, 10

plot.MedianPolishM, 7, 8, 10

removetrendMPst, 11

splineMPST, 13

VRes, 14