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` *Construct Spatio - temporal regular data.*

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

Martínez, W. A., Melo, C. E., & Melo, O. O. (2017). *Median Polish Kriging for space–time analysis of precipitation* Spatial Statistics, 19, 1-20. [link]

Berke, O. (2001). *Modified median polish kriging and its application to the wolfcamp - aquifer data.* Environmetrics, 12(8):731-748.[link]

Examples

```
## 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

```
data(DemMeta)
```

Format

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

Source

http://www.jspacesystems.or.jp/ersdac/GDEM/E/index.html

Examples

```
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

```
data(HZRMeta)
```

Format

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

Source

http://www.arcgis.com/home/item.html?id=103b63dcc9f448acbd63f22b728b1a02

Examples

```
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

```
krige0STlocalMP(data,newdata,p,model,k,stAni)
```

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.

`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

Martínez, W. A., Melo, C. E., & Melo, O. O. (2017). *Median Polish Kriging for space–time analysis of precipitation* Spatial Statistics, 19, 1-20. [link]

Pebesma, E.J. (2004). *Multivariable geostatistics in S: the gstat package*. Computers & Geosciences, 30: 683-691 [link]

Pebesma, E.J. (2012). *spacetime: Spatio-Temporal Data in R*. Journal of Statistical Software, 51(7), 1-30.[link]

Examples

```
library(spacetime)
library(sp)
library(gstat)
library(zoo)
library(maptools)
data(Metadb)
#records of the precipitation monthly 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(sqrt(0.5+Metadb[,-c(1:4)]),time,pts=Metadb[,2:4],Delta=c(7,6,5))
residual<-removetrendMPst(MPST,eps=0.01, maxiter=2)
rain.loc<-Metadb[,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")
```

```
Grid_pred <- STF(sp=SpatialPoints(Gridxyz,CRS(proj4string(DemMeta))), time=time[c(18,19)])

#Product - sum covariance model generalized
p=c(2,12.98,13899.95,3.44,14.95,1.84,3.92,-0.07)
CS = function(h,p){p[2]*exp(-h/p[3])}
CT = function(u,p){p[4]*exp(-u/p[5])+ p[6]*cos(pi*u/180)+p[7]*(1-abs(sin(pi*u/180)))}
CST<-function(h,u,p){0.084*CT(u,p)+ 0.32*CS(h,p)+0.07*CT(u,p)*CS(h,p)}
data(VRes)
stAni<-estiStAni(VRes, interval=c(10, 100))

PredictValue<-krige0STlocalMP(data=rain_residual,newdata=Grid_pred,p,model=CST,k=10,stAni)
IDs = paste("ID",1)
mydata = data.frame(PredictValue[,5], ID=IDs)
wind.ST1 = STFDF(SpatialPixels(Gridxy),time[c(18,19)],mydata)
stplot(wind.ST1,col.regions=bpy.colors(40),par.strip.text = list(cex=0.7)
       ,main="Kriging ordinary residuals: Prediction surface")
```

MedianPolishM *Median polish multidimensional.*

Description

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

Martínez, W. A., Melo, C. E., & Melo, O. O. (2017). *Median Polish Kriging for space–time analysis of precipitation* Spatial Statistics, 19, 1-20. [link]

Hoaglin, D. C., Mosteller, F., & Tukey, J. W. (Eds.). (2011). *Exploring data tables, trends, and shapes* (Vol. 101). John Wiley & Sons.[link]

`MedianPolishM.ConstructMPst` *Median polish multidimensional.*

Description

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

Usage

```
## 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 μ is an overall mean, α_a is the a-th row effect, β_b is the effect b-th column effect, ξ_c is the c-th layer effect, τ_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

Martínez, W. A., Melo, C. E., & Melo, O. O. (2017). *Median Polish Kriging for space–time analysis of precipitation* Spatial Statistics, 19, 1-20. [link]

Hoaglin, D. C., Mosteller, F., & Tukey, J. W. (Eds.). (2011). *Exploring data tables, trends, and shapes* (Vol. 101). John Wiley & Sons.[link]

`MedianPolishM.default` *Median polish multidimensional.*

Description

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

Usage

```
## 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 $dim_1, \cdots, dim_n$, 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

Hoaglin, D. C., Mosteller, F., & Tukey, J. W. (Eds.). (2011). *Exploring data tables, trends, and shapes* (Vol. 101). John Wiley & Sons.[link]

Examples

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

`Metadb` *Monthly precipitation Meta.*

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)
```

Mpplot *Traces of the space.*

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(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))
Mpplot(MPST)
```

plot.MedianPolishM *Plot Median polish multidimensional.*

Description

Plot the effects of an additive model for multidimensional array, using Tukey's median polish procedure.

Usage

```
## 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 ConstructutMPst, this method has a specific graphic for data with space - time variability.

References

Martínez, W. A., Melo, C. E., & Melo, O. O. (2017). *Median Polish Kriging for space–time analysis of precipitation* Spatial Statistics, 19, 1-20. [link]

Hoaglin, D. C., Mosteller, F., & Tukey, J. W. (Eds.). (2011). *Exploring data tables, trends, and shapes* (Vol. 101). John Wiley & Sons.[link]

Examples

```
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 throught median polish.

Usage

```
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, Martinez et al.(2017) used a structure of four dimensions to apply median polish tecnique.

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

$$Y(\mathbf{s}_{abc}, t) = \mu_y(\mathbf{s}_{abc}, t) + \delta_{abct}$$

where

$$\mu_y(\mathbf{s}_{abc}, t) = \mu + \alpha_a + \beta_b + \xi_c + \tau_t$$

and δ_{abct} is a fluctuation arising from natural variability and from the measurement process. Additionally, μ is an overall mean, α_a is the a-th row effect, β_b is the effect b-th column effect, ξ_c is the c-th layer effect and τ_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

Martínez, W. A., Melo, C. E., & Melo, O. O. (2017). *Median Polish Kriging for space–time analysis of precipitation* Spatial Statistics, 19, 1-20. [link]

Berke, O. (2001). *Modified median polish kriging and its application to the wolfcamp - aquifer data.* Environmetrics, 12(8):731-748.[link]

Examples

```
## 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))
```

```
residual<-removetrendMPst(MPST,eps=0.01, maxiter=2)
## End(Not run)
```

---

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 scenary 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

Martínez, W. A., Melo, C. E., & Melo, O. O. (2017). *Median Polish Kriging for space–time analysis of precipitation* Spatial Statistics, 19, 1-20. [link]

Berke, O. (2001). *Modified median polish kriging and its application to the wolfcamp - aquifer data.* Environmetrics, 12(8):731-748.[link]

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]
```

```
}
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<-MedianPolishM(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

Martínez, W. A., Melo, C. E., & Melo, O. O. (2017). *Median Polish Kriging for space–time analysis of precipitation* Spatial Statistics, 19, 1-20. [link]

Pebesma, E.J. (2004). *Multivariable geostatistics in S: the gstat package*. Computers & Geosciences, 30: 683-691 [link]

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$gamma - mod
  if (trace)
    print(c(p, MSE = mean(resid^2)))
  mean(resid^2)
}
ModSpatial = function(h,p){p[2]*(1-exp(-h/p[3]))}
ModTemporal = function(u,p){p[4]*(1-exp(-u/p[5]))+ p[6]*(1-cos(pi*u/180))+
p[7]*abs(sin(pi*u/180))}
VariogST=function(h,u,p)
{ModTemporal(u,p)+ModSpatial(h,p)+p[8]*ModTemporal(u,p)*ModSpatial(h,p)}
#Parametros Iniciales
p1<-c(2,14.5,13900,5.9,29,1.55,3.7,-0.07)
pars.st = optim(p1, FitPar_st, method =  "BFGS",
                gfn = VariogST, v = VRes, hessian=TRUE)
fit_Variog_ST<-VRes
fit_Variog_ST$gamma<-VariogST(VRes$spacelag, VRes$timelag, pars.st$par)
plot(fit_Variog_ST)
```

Index