

Package ‘ipdw’

July 2, 2014

Title ipdw: Interpolation by Inverse Path Distance Weighting

Description Interpolation of geo-referenced point data by Inverse Path Distance Weighting

Version 0.2-1

Author Joseph Stachelek <jstachel@sfwmd.gov>

Maintainer Joseph Stachelek <jstachel@sfwmd.gov>

Depends R (>= 3.0.2),gdistance

Imports raster

Suggests geoR,gstat,gdata,spatstat

License GPL (>= 2)

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2014-06-16 16:40:40

R topics documented:

costrasterGen	2
errorGen	3
ipdw	4
ipdwInterp	5
pathdistGen	6
Index	7

costrasterGen *Generate a cost Raster*

Description

generate a cost raster from an object of class "SpatialPolygons"

Usage

```
costrasterGen(xymat, polys, extent = "polys", projstr)
```

Arguments

xymat	Matrix of coordinates
polys	SpatialPolygons object
extent	Define extent based on extent of xymat (pnts) or polys (polys). Default is polys.
projstr	proj4 string defining the inherent projection

Value

RasterLayer

Author(s)

Joseph Stachelek

Examples

```
Sr1<-Polygon(cbind(c(0,0,2,2,0),c(0,4,4,0,0)))
Sr2<-Polygon(cbind(c(2,2,4,4,2),c(0,2,2,0,0)))
Srs1<-Polygons(list(Sr1), "s1")
Srs2<-Polygons(list(Sr2), "s2")
polys<-SpatialPolygons(list(Srs1,Srs2), 1:2)

xymat<-matrix(3,3,nrow=1,ncol=2)
costras<-costrasterGen(xymat,polys,projstr=NULL)
plot(costras)
points(xymat)
```

errorGen	<i>Generate interpolation error stats from validation datasets</i>
----------	--

Description

Generate interpolation error stats from validation datasets

Usage

```
errorGen(finalraster, validation.spdf, validation.data, plot = FALSE,  
         title = "")
```

Arguments

finalraster	RasterLayer object
validation.spdf	SpatialPointsDataFrame
validation.data	data.frame
plot	logical. Plot comparison?
title	Plot labels

Value

List of error statistics

Author(s)

Joseph Stachelek

Examples

```
validation.data<-data.frame(rnorm(10,mean=0.2,sd=1))  
names(validation.data)<-c("validation")  
validation.spdf<-validation.data  
validation.data<-as.numeric(unlist(validation.data))  
xy<-data.frame(x=c(0:9),y=rep(1,10))  
coordinates(validation.spdf)<-xy  
  
m<-matrix(NA,1,10)  
out.ras<-raster(m,xmn=0,xmx=ncol(m),ymn=0,ymx=nrow(m))  
out.ras[]<-validation.data+rnorm(ncell(out.ras),mean=0.01,sd=0.2)  
  
valid.stats<-errorGen(out.ras,validation.spdf,validation.data,plot=TRUE,title="Validation Plot")  
valid.stats
```

ipdw

Inverse Path Distance Weighting

Description

Inverse Path Distance Weighting

Usage

```
ipdw(spdf, costras, range, paramlist, overlapped = FALSE,  
     yearmon = "default", removefile = TRUE, step = 16)
```

Arguments

spdf	SpatialPointsDataFrame object
costras	RasterLayer. Cost raster
range	numeric. Range of interpolation neighborhood
paramlist	character. String representing parameter names
overlapped	logical. Default is FALSE, specify TRUE if some points lie on top of barriers
yearmon	character. String specifying the name of the spdf
removefile	logical. Remove files after processing?
step	numeric. Number of sub loops to manage memory during raster processing.

Details

This is a high level function that interpolates a SpatialPointsDataFrame object in a single pass.

Value

RasterStack

Author(s)

Joseph Stachelek

Examples

```
#see vignette
```

ipdwInterp

*Inverse Distance Weighting with custom distances***Description**

Inverse Distance Weighting with custom distances

Usage

```
ipdwInterp(spdf, rstack, paramlist, overlapped = FALSE, yearmon = "default",
  removefile = TRUE)
```

Arguments

spdf	SpatialPointsDataFrame object
rstack	RasterStack of path distances
paramlist	character. String representing parameter names
overlapped	logical. Default is FALSE, specify TRUE if some points lie on top of barriers
yearmon	character. String specifying the name of the spdf
removefile	logical. Remove files after processing?

Value

RasterLayer

Author(s)

Joseph Stachelek

Examples

```
spdf<-data.frame(rnorm(2))
xy<-data.frame(x=c(4,2),y=c(8,4))
coordinates(spdf)<-xy
m<-matrix(NA,10,10)
costras<-raster(m,xmn=0, xmx=ncol(m), ymn=0, ymx=nrow(m))
#introduce spatial gradient
costras[]<-runif(ncell(costras),min=1,max=10)
for(i in 1:nrow(costras)){
  costras[i,]<-costras[i,]+i
  costras[,i]<-costras[,i]+i
}

rstack<-pathdistGen(spdf,costras,100)
final.raster<-ipdwInterp(spdf,rstack,paramlist=c("rnorm.2. "))
plot(final.raster)
```

pathdistGen	<i>Generate a stack of path distance raster objects</i>
-------------	---

Description

Generate a stack of path distance raster objects

Usage

```
pathdistGen(spdf, costras, range, step = 16, yearmon = "default")
```

Arguments

spdf	SpatialPointsDataFrame object
costras	RasterLayer cost raster
range	numeric. Range of interpolation neighborhood
step	numeric. Number of sub loops to manage memory during raster processing.
yearmon	character. String specifying the name of the spdf

Value

RasterStack object of path distances

Author(s)

Joseph Stachelek

Examples

```
spdf<-data.frame(rnorm(2))
xy<-data.frame(x=c(4,2),y=c(8,4))
coordinates(spdf)<-xy

m<-matrix(NA,10,10)
costras<-raster(m,xmn=0,xmx=ncol(m),ymn=0,ymx=nrow(m))
costras[]<-runif(ncell(costras),min=1,max=10)
#introduce spatial gradient
for(i in 1:nrow(costras)){
  costras[i,]<-costras[i,]+i
  costras[,i]<-costras[,i]+i
}

rstack<-pathdistGen(spdf,costras,100)
```

Index

costrasterGen, [2](#)

errorGen, [3](#)

ipdw, [4](#)

ipdwInterp, [5](#)

pathdistGen, [6](#)