Package 'SurfaceTortoise'

October 2, 2020

Type Package

Title Find Optimal Sampling Locations Based on Spatial Covariate(s)

Version 1.0.2

Author Kristin Piikki, Mats Söderström & John Mutua

Maintainer Kristin Piikki <kristin.piikki@slu.se>

Description Create sampling designs using the surface reconstruction algorithm.
Original method by: Olsson, D. 2002. A method to optimize soil sampling from
ancillary data. Poster presenterad at: NJF seminar no. 336,
Implementation of Precision Farming in Practical Agriculture, 10-12
June 2002, Skara, Sweden.

Depends R (>= 3.4.4)

Imports raster, gstat, rgeos, sp

Suggests roxygen2

License MIT + file LICENSE

URL https://CRAN.R-project.org/package=SurfaceTortoise

BugReports https://github.com/soilmapper/SurfaceTortoise/issues/

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2020-10-02 08:32:08 UTC

R topics documented:

tortoise *SurfaceTortoise*

Description

Optimizing spatial sampling using the Surface Tortoise algoritm. Grid sampling and random sampling are also available. All three sampling designs can optionally be stratified by a square grid to ensure spatial coverage.

Usage

```
tortoise(
  x = NULL,
  y = NULL,
  method = "directed",
  edge = 0,
  strat_size = NULL,
  min_dist = 0,
  p_idw = 2,
  nmax_idw = 8,
  resolution = NULL,
  filter = 1,
  stop_n = NULL,
  stop_dens = 1,
  plot_results = F
)
```

Arguments

x	Raster dataset. Required for method = directed. The raster must have a defined coordinate system and must be of class numeric. If x is a raster stack or raster brick, the first principal compinent of the multiple layers will be used for sampling. If the raster dataset has a single layer, it will be used as is.
y	SpatialPolygonsDataframe delineating the area to be sampled. Required for method = 'grid' and method = 'random'. Optional for method = 'directed. The SpatialPolygonsDataframemust must have a defined coordinate system and, if a raster is provided, the coordinate system shall be the same as for the raster. If x and y are not completely overlapping, their intersection will be sampled.
method	Sampling method: 'directed' = directed sampling (SurfaceTortoise algorithm), 'grid' = regular sampling (center points of strata) and 'random' = random points. Default is 'directed'
edge	A number. Buffer zone (metre) inside the sampled area border, where sampling is prohibited. Optional.
strat_size	A number. Cell side (metre) of a square stratification grid. Optional. #' If both strat_size and stop_n are specified. stop_n overruns this argument #' with an adjusted strat_size. If strat_size is not specified. The sampling will be done

without stratification. If strat_size = 0, stratification size will be comuted from the number of samples. Negative values are not allowed.

`min_dist` A positive.number. Minimum distance allowed between samples. Valid for the 'random' and the 'directed' methods.

`p_idw` An integer. Power exponent used for idw-interpolation (method = 'directed'). Default is 2.

`nmax_idw` An integer. Number of neighbouring samples used for idw-interpolation (method = 'directed'). Default is 8.

`resolution` An number. If provided, the raster data vill be resampled to this resolution. Optional.

`filter` An integer. Side of the square window (number of raster cells, original resolution) used for mean filtering of the raster. Default = 1 (no filtering)

`stop_n` An integer. The number of samples to place. If not provided, it will be conuted from the numbers of strata generated from the specificed stratication size (argument strat_size) and the number of samples to place per stratum (argument stop_dens).

`stop_dens` An integer. The number of samples to place in each stratum. Does not apply for method = 'grid' (always stop_dens = 1) and not for non-stratified sampling. Default is 1.

`plot_results` Logical. Shall results be plotted? Default is FALSE.

Details

The Surface Tortoise algorithm for directed sampling uses a raster dataset to find optimal sample locations. The sampling strategy is based on the principle that an interpolation of the samples should be as similar as possible to the guide raster. When sample locations are identified, first the center point of the raster cell with the maximum deviation from the covariate raster mean is sampled. Then the raster cell with the maximum deviation from the first sampled raster cell is sampled. From then on, the values of the sampled raster cells are interpolated by inverse distance weighting (idw) and the center point of the raster cell with the largest absolute difference to the guide raster (error) is sampled. A new idw interpolation is made and a new cell is sampled. This is repeated is reached.The sampling can be stratified by a square grid. When a sample has been placed in a stratum, no more samples will be placed in that stratum again until all other strata have been sampled. The likelihood for a clipped stratum, e.g. at the edge of the area to be sampled, is equal to the area of that stratum divided by the area of a full stratum.

The optional raster processing steps: (is done) is carried out in the folowing order: 1) mean filtering (argument: filter) 2) resampling to specified resolution (argument: resolution), 3) computation of first pricipal component (if x is a rastr stack or raster brick with multiple layers).

Value

A list with 1) sampled_raster = the sampled raster (only if method = 'directed') 2) samples = a spatialPointsDataFrame with sample locations 3) sampled_area = a SpatialPolygonsDataFrame with a polygon for the sampled area. 4) stratification = a a SpatialPolygonsDataFrame with the stratification polygons. 5) feedback= a dataframe with generated text messages.

Author(s)

Kristin Piikki, Mats Söderström & John Mutua, <kristin.piikki@slu.se> <kristin.piikki@slu.se>

References

Olsson, D. 2002. A method to optimize soil sampling from ancillary data. Poster presenterad at: NJF seminar no. 336, Implementation of Precision Farming in Practical Agriculture, 10-12 June 2002, Skara, Sweden.

Examples

```
#create a boundary polygond for the area to be sampled
coords<- c(1, 4, 3, 4, 3, 5, 1, 5)
coords <-matrix(data=coords, ncol=2, byrow=TRUE) #coordinates
prj<-'+init=epsg:3857' #projection
poly<-list(sp::Polygon(coords)) #polygon
poly<-list(sp::Polygons(poly,'id')) #polygon
poly <- sp::SpatialPolygons(poly, proj4string=sp::CRS(prj)) #polygon
#do grid sampling
grid<-tortoise(y=poly,method='grid',edge=0.1,strat_size=0.2,
               min_dist=10,plot_results=TRUE)
```

Index