Package 'CanHeMonR.MaxEnt'

August 14, 2018

Type Package

Title Application of MaxEnt for canopy health monitoring through airbone image analysis.

Version 2.0

Date 2018-03-15

Maintainer Laura Martinez <lmartisa@gmail.com>

Author Pieter Beck <Pieter.BECK@ec.europa.eu>

Description MaxEnt model for the detection of declining trees in Portugal through high definition multispectral airbone image.

Depends R (>= 3.4)

Imports raster, doParallel, dismo, foreach, maptools, rgeos, rJava, DiagrammeR, sp, sf, units, dplyr, rgdal

License GPL (>= 3)

LazyData TRUE

RoxygenNote 6.0.1

R topics documented:

`Buffer` *Buffered point method*

Description

Buffer out a point with an specific size and then compare the classes Pb and fals_pos to see if we can separate them in the post-process stage. It also creates the graphs to check this information. A boxplot graph with all the quantiles and a graph to see how much fasle_pos can be extracted vs Pb retrieved.

Usage

```
Buffer(pbtiles, visualpoints, polyspath, rasterpath, prefix, outfile)
```

Arguments

`pbtiles`	txt file with the tiles names that you want to analizse
`visualpoints`	shapefile where all the PB points are
`polyspath`	Path where the shapefiles to analise
`rasterpath`	Path to the raster files associates with the shapefiles
`prefix`	Prefix of the Maxent shapefiles, if there is a prefix
`outfile`	Path where you want to save the stats and graphs generated

Value

An rdsdata dataframe with the base info and the plots

`Calibrate_model` *Calibrate vegetation distribution models*

Description

For each class in .shp polygon file, calibrate a distribution model using a raster brick as predictors.

Usage

```
Calibrate_model(vuln_classes = "ALL", training_path, model_outp_dir, name,
  stadistics = FALSE, myargs = NULL, model_type, stadisticspath = NULL)
```

Arguments

vuln_classes	A character vector of the classes you want to model. The should be presented in the column 'class' of training_df. Default 'ALL'
training_path	Path to the rdsdata that contains the data.frame, with in the column 'pres' 1/0 to indicate presence absence, then covariate columns, and a colum 'class' groupin grows by the land-cover class the data was sampled for. This df is typically generated by sample_points.r
model_outp_dir	Path and filename prefix to save the model objects
name	Character. Name that you want to give to the serialized object with the model
stadistics	Boolean. If Ture stadistics of the model will be done and save. Take into account that it can take several time. Default FALSE
myargs	List. Arguments to pass to the maxent model, in the following format. Example: myargs <- c("noautofeature", "nohinge", "nothreshold", "noproduct","nolinear"). Default NULL
model_type	Character. Type of model that we wnat to use to predict types: raw - logistic - cloglog
stadisticspath	Path where you want to save the stadistics of the model. Default NULL

Value

Serialized object with class-specific distribution models, using a data frame created from training points and covariate images

See Also

For more possible arguments for MaxEnt see: https://groups.google.com/forum/#!msg/Maxent/yRBlvZ1_9rQ/Fj8Two0l

Depends on: sick_tree_errors.r

Examples

```
## Not run:
tt <- calibrate_model(vuln_classes = list(c('Pb')),
                      training_path = '/DATA/Results/RunSickTree_Output/ADS/Test/MaxentMo
                      model_outp_dir = '/DATA/Results/RunSickTree_Output/ADS/Final_Iter1-
                      name='samp_NA_buffer_logistic',
                      stadistics = TRUE,
                      myargs = c("noautofeature", "nohinge"),
                      model_type = 'cloglog',
                      stadisticspath = '/DATA/Results/RunSickTree_Output/ADS/Test/MaxentM

## End(Not run)
```

`Classify_maxent_output_based_on_error_stats` *Classify maxent output to binary presence absence maps*

Description

Classify maxent output to binary presence absence maps based on thresholds of pixel-level probability, expected minimum crown size, and clump size, finally outputs a polygon shapefile.

Usage

```
Classify_maxent_output_based_on_error_stats(r_pred_dir, tile = "ALL",
  max_permittable_cutoff_final = max_permittable_cutoff_final,
  max_permittable_cutoff_npix_final = max_permittable_cutoff_npix_final,
  max_area, res, outp_dir)
```

Arguments

`r_pred_dir`	A directory where binary .tifs predicting presence as 1 and absence as 0 can be found for multiple tiles
`tile`	Character vector. Names of tile(s) to run. 'ALL will run all tiles in r_pred_dir. Default is 'ALL'
`max_permittable_cutoff_final`	Cutoff-level for Maxent probability output. Typically produced by cut_off_selection.r
`max_permittable_cutoff_npix_final`	Cutoff-level for the nr of pixels (clumpsize) of presence that a circles of radius radius needs to contain to be maintained as presence. Typically produced by error_summaries.R
`max_area`	The maximum are aof a polygon, all the polygons above this area will be filtered
`res`	The resolution of the raster
`outp_dir`	The directory where you want to save the output shapefiles.

Value

A shapefile for each binarized raster file

See Also

Depends on: `cut_off_selection.r`

`COSoutput` *Intersect COS file*

Description

Intersect the shapefile resulting from the last step of Maxent with the COS file. It also can be applied to any intersection between to shapefiles

Usage

```
COSoutput(outpath, shppath, COSpath, prefix = NULL)
```

Arguments

`outpath`	Path wwhere you want to save the resultin shapefiles
`shppath`	Path where the Maxent shapefiles are
`COSpath`	Full path plus name of the COS file
`prefix`	Prefix of the Maxent shapefiles, if there is a prefix

Value

Several shapefies with the points that intersect with the COS file

`Createdataset` *Outputdataset*

Description

Create a dataset of the visual inspected shapefile and save it.

Usage

```
Createdataset(shp_insp, listrasters, shp_poly_path, prefix, savepath)
```

Arguments

`shp_insp`	Path to the visual inspected shapefile
`listrasters`	csv file with the silt of locations(rasternames) you want to study.
`prefix`	Prefix that you want to put to the plots in order to differenciate them. Default = NULL
`savepath`	Path where you want to save the dataframe

Value

A dataframe with all the values of the validated shapefile. Used to create the stadistics of the model

`GetPoints` *Sample points of presence points and background points and save them in a serialized file*

Description

Sample points of presence points and background points and save them in a serialized file.

Usage

```
GetPoints(tile_i, all_tifs, field_name, ninputs_tile, randompt, prob_tifs, Pols,
  vuln_classes, abs_samp, tile_dat)
```

Arguments

`tile_i`	Passed by the algorithm, name of the tile to run
`all_tifs`	Passed by the algorithm, list of all the tiles to run
`field_name`	Character. The field in AOI.filename that contains the vuln_classes
`ninputs_tile`	Integer. Number of inputs that we have fore each tile, including the tile, for exemple number of textures
`randompt`	Boolean. if True random points will be added in the tiles that doesn't have any visual point. Default TRUE
`prob_tifs`	Boolean (FALSE) if not wanted, Directory if wanted. tifs of each raster to run with 0 vaue for the areas that we don't want to sample and 1 fore the ones that we want to sample. Default FALSE
`Pols`	Passed by the algorithm, PolygonDataframe Object with all the visual points assesed
`vuln_classes`	A list of the classes you want to model The list can contain one or more vectors characters. Each element of the vector represents a seperate vegetation class and response variable for the model and the vector elements are synonyms used to describe that class The fist place in each vector will be used in the output name used to store the calibrated model, so it should not contain spaces. The other places should appear as attributes in the field 'field_name' of Pols
`abs_samp`	Integer. How many 'absence' pixels should be randomly selected from eah tile to train the model. Default is 100.
`tile_dat`	A dataframe with the points of the prevoius iteration, if the previous iteration is the initial one, an empy dataframe will be passed

Value

A dataframe with the points done for the tile selected

Intersection *Point in polygon*

Description

Saves in a data frame all the intersections between a polygon shapefile and a point shapefile. Used for the Crown delineation method

Usage

```
Intersection(inpath, outpath, pointpath)
```

Arguments

inpath	Path where the polygons shapefiles are
outpath	Path where you weant to save the resulting point shapefile
pointpath	Shapefile where all the PB points are

Value

A rdsdata dataframe called intersections

MaxCutoff *Maximum cutoff*

Description

Selects, given a maximum error, the cut-off probability and the minimum amout of pixels that a tree have to have. This information is important to do the final polygonization of the raster images outputted by MaxEnt. After that it binarize the rasters and transformates then into a polygon shapefile.

Usage

```
MaxCutoff(input_dir, plots, stadisticspath, prefix, max_error, cla, r_pred_dir,
  tile = "ALL", max_area, res, outp_dir)
```

Arguments

input_dir	Full path where the serialized objects with the threshold informtion are
plots	Boolean. If TRUE plots will be created. Default FALSE Plot-1: Plot-2: error in the tree dectection Plot-3: Distribution of the number of pixels for the trees correctly decteted
stadisticspath	Full path where the plots will be saved
prefix	Prefix that you want to put to the plots in order to differenciate them. Default = NULl
max_error	Integer. Maximum error you want to select

cla	Name of the attribute that you want to do the test on. Ex: 'Pb'
r_pred_dir	Path where the rasters outputted by Maxent to binarize and poligonize are.
tile	Path to a txtfile with the names of the tile you want to execute If you want to run all the images of r_pred_dir, leave it empty. Default is 'ALL'
max_area	Maximum area in cm. If the area detected on the image is above this value it will be erase
res	Pixel resolution of the image. Ex: 0.30
outp_dir	Path where you want to save the shapefiles of the binarized raster

Value

Polygon shapefile

NDVI *NDVI by moving window*

Description

Calculates the NDVI given a moving window size.

Usage

```
NDVI(rast, sizemv)
```

Arguments

rast	raster to calculate the ndvi
sizemv	Size desired for the moving window

Value

the resulting NDVI raster

---

ortho2012 *Filter shapefile with th ADS 2012 evolution on NDVI*

Description

Filter shapefile with th ADS 2012 evolution on NDVI.

Usage

```
ortho2012(filtpath, shppath, outpath, prefix)
```

Arguments

filtpath	path tho the shapefiles that will act as filter
shppath	shapefile that you want to filter
outpath	Path where you want to save the shp generated
prefix	Prefix of the Maxent shapefiles, if there is a prefix

Value

a shapefile filtered

`Par` *Parallel enviroment*

Description

Set up the parallel enviroment if needed.

Usage

```
Par(nWorkers, data_outp_dir, data_outp_name, parallel)
```

Arguments

`nWorkers`	Number of threads you want to use the maximum is Maxcores-1
`data_outp_dir`	Path where you want to save the logfile
`data_outp_name`	Name of the log file
`parallel`	Boolean. True for parallel performance

Value

The registration of parallel flag

`Pointshp` *Polygont to point*

Description

Creates a unique point shapefiles from several polygons shapefiles.

Usage

```
Pointshp(inpath, outpath,
  chunks = "/HDD/visual_interpretation/visual_interpretation_ADS/ADS100_Chunks.s
  name_chunk)
```

Arguments

`inpath`	Path where the polygons shapefiles are
`outpath`	Path where you weant to save the resulting point shapefile
`chunks`	Path and name of the shapefile with the chunks
`name_chunk`	Name of the chunk that you want to process

Value

A unique shapefile for the centroids of the polygons

Poligonize *Poligonize Raster*

Description

Poligonize a raster with values between 0 and 1.

Usage

```
Poligonize(rast, rastpath)
```

Arguments

rast	Path to the raster to polyginze

Value

A SpatialDataFrame object

ReadThresh *Read Threshold*

Description

Read the thresholds calculated in: and create a dataframe with the information.

Usage

```
ReadThresh(input_dir)
```

Arguments

input_dir	Full path where the serialized objects with the threshold informtion are sick_tree_errors.r

Value

A dataframe contaning all the infor of the different threshold for the selected tiles

Roadoutput *Intersect road CODfile*

Description

Intersect the shapefile with the COS and roads file to erase the points that intersects with the polygons.

Usage

```
Roadoutput(outpath, shppath, RoadCOSpath, prefix = NULL)
```

Arguments

outpath	Path wwhere you want to save the resultin shapefiles.
shppath	Path where the Maxent shapefiles are
RoadCOSpath	Full path plus name of the road file from OSM
prefix	Prefix of the Maxent shapefiles, if there is a prefix

Value

Several shapefies with the points that don't intersect with the Road file

Run_model *Run a saved MaxEnt model in predictive mode on a tile of image data*

Description

Run a saved MaxEnt model on a the image data selected.

Usage

```
Run_model(predictor_dir, text_train_dir, MaxEntmodel_dir, fname_MaxEntmodel_r,
  output_dir, rastername, model_type, EOS)
```

Arguments

predictor_dir
: Path where predictor layers are held, rasters. If EOS = FALSE predictor_dir is a path if EOS = True predictor_dir is the path plus the image to predict

text_train_dir
: Path where .tifs of the textures associated with r_train_dir. It is really important to avoid errors on the execution to pass the same numer of textures per tile as in the MaxEnt model used

MaxEntmodel_dir
: Path where the MaxEnt model file is held

fname_MaxEntmodel_r
: Filename of the MaxEnt model saved in rdsdata format

output_dir	Path to write the output to
rastername	Character. Prefix to give the outputed raster image, for control versions
model_type	Character. Type of model of maxent you want to use: raw, logistic or cloglog
EOS	If EOS true the for loop will be avoided if False will work with a for loop. Default FALSE

Value

A raster image for each tile with the probabilities or cummulative probabilities of presence for each class

See Also

Depends on: calibrate_model.r

Examples

```
## Not run:
Run_model(predictor_dir = "/H03_CANHEMON/Imagery/Portugal/ADS100/ortophotos_06032017/geot
                          text_train_dir <-'/home/martlur/Documents/TexturesAds/',
                          MaxEntmodel_dir = "/home/martlur/Documents/Dockers/docker6EOS
                          fname_MaxEntmodel_r = "samp10000_Pb.rdsdata",
                          output_dir = "/DATA/Results/Rcode/OutputRunSickTree",
                          rastername = "samp1000_",
                          model_type = 'cloglog',
                          loop = FALSE)

## End(Not run)
```

Sample_points *Sample training data for image classification from multiple image tiles*

Description

For each class in .shp polygon file, Sample training data for image classification from multiple image tiles using their raster bricks as predictors.

Usage

```
Sample_points(r_train_dir, text_train_dir, tile = "ALL", text = "ALL",
  prob_tifs = FALSE, vuln_classes, training_pol_filename, field_name,
  ninputs_tile, data_outp_dir, abs_samp = 100, parallel = F, nWorkers = 4,
  data_outp_name, randompt = TRUE, EOS = FALSE)
```

Arguments

r_train_dir	A directory where .tifs for training can be found for multiple tiles
text_train_dir	A directory where .tifs of the textures associated with r_train_dir
tile	Character vector or CVS file. Names of tile(s) to run a cvs file. 'ALL' will run all tiles in r_train_dir. Default is 'ALL'

`text` Character vector or CVS file. Names of text(s) to run in a cvs file. 'ALL' will run all tiles in text_train_dir. Default is 'ALL'

`prob_tifs` Boolean (FALSE) if not wanted, Directory if wanted. tifs of each raster to run with 0 vaue for the areas that we don't want to sample and 1 fore the ones that we want to sample. Default FALSE

`vuln_classes` A list of the classes you want to model The list can contain one or more vectors. Each vector represents a seperate vegetation class and response variable for the model and the vector elements are synonyms used to describe that class The fist place in each vector will be used in the output name used to store the calibrated model, so it should not contain spaces. The other places should appear as attributes in the field 'field_name' of Pols

`training_pol_filename` Full path to the vector file (SpatialPointsDataFrame) of which one field contains the vuln.classes

`field_name` Character. The field in AOI.filename that contains the vuln_classes

`ninputs_tile` Integer. Number of inputs that we have fore each tile, including the tile, for exemple number of textures

`data_outp_dir` The folder where you want to save the sampled data

`abs_samp` Integer. How many 'absence' pixels should be randomly selected from eah tile to train the model. Default is 100.

`parallel` Boolean. Should the code be run in parallel using the doParallel package? Default is FALSE

`nWorkers` Integer. If running the ocde in parallel, how many workers should be used? Default is 4

`data_outp_name` Character. Name of the data to output

`randompt` Boolean. if True random points will be added in the tiles that doesn't have any visual point. Default TRUE

`EOS` If EOS true the for loop will be avoided if False will work with a for loop. Default FALSE

Value

Saves a serialize object with the list with class-specific data frames of which the first column is the presence-absence response that can be used to train distribution model.

`Sick_tree_errors` *Sick tree error calculator*

Description

Calculate errors in automated detection of declining trees using visual inspection data as reference.

Usage

```
Sick_tree_errors(r_pred_dir, tile = "ALL", prefix, vuln_classes,
  training_pnt_filename, radius = 2, field_name, abs_samp = 100,
  minthresh = 0, maxthresh = 1, stepthresh = 0.05, parallel = F,
  nWorkers = 4, data_outp_dir = NULL)
```

Arguments

r_pred_dir	A directory where binary .tifs predicting presence as 1 and absence as 0 can be found for multiple tiles
tile	path to a txtfile with the names of the tile you want to execute If you want to run all the images of r_pred_dir, leave it empty. Default is 'ALL'
prefix	Prefix that you want to add to the output file. EX: NorthPortugal NorthPortugal-> -sicktree_performance_dfs.rdsdata
vuln_classes	A list of the classes you want to model. The list can contain one or more vectors. Each vector represents a seperate vegetation class and response variable for the model and the vector elements are synonyms used to describe that class. The fist place in each vector will be used in the output name used to store the calibrated model, so it should not contain spaces. The other places should appear as attributes in the field 'field_name' of pnts.
training_pnt_filename	Full path to the SpatialPointsDataFrame of which one field contains the vuln_classes
radius	The radius within which a presence point must be found for it to be considered 'correct'. Default is True
field_name	Character. The field in AOI.filename that contains the vuln_classes
abs_samp	How many 'absence' pixels should be randomly selected from each tile to evaluate the absences? Default is 100.
minthresh	The minimum cut-off number you want to use, Default = 0
maxthresh	The maximum cut-off number you want to use, Default = 1
stepthresh	The step used to create de sequences of threshold from the minthresh to the max thresh. Default = 0.05
parallel	Should the code be run in parallel using the doParallel package? Default is FALSE.
nWorkers	If running the ocde in parallel, how many workers should be used? Default is 4.
data_outp_dir	The folder to save the sampled data to. No data is saved is data_outp_dir is NULL. Default is NULL.

Value

A data frame with commission and ommission errors and sample sizes of presence and absence

---

Tiles *Treatment of tiles*

Description

Check if you want to read all the tiles from a folder or just the files needed via csv.

Usage

```
Tiles(r_train_dir, tile, text_train_dir, text)
```

Arguments

r_train_dir	Path where the tiles are located
tile	Or 'ALL' for reading all the files, or the path where the csv is located with the name of the rasters
text_train_dir	Path where the textures are located
text	Or 'ALL' for reading all the files, or the path where the csv is located with the sufix of the textures wanted

Value

The list of files to process

Validationplots *Validation plots*

Description

Creation of the plots related to the validation of the predictions of MaxEnt.

Usage

```
Validationplots(pathrds, validationweb = TRUE, savepath, prefix = NULL)
```

Arguments

pathrds	Path to the serialized object with the information of the visual inspected area
savepath	Path where you want to save the dataframe
prefix	Prefix that you want to put to the plots in order to differenciate them. Default = NULL

Value

A series of plots of Maxent results, true positives and False positives, Accurancy tratio, error, etc.

Index