Package ‘dissever’

October 13, 2022

Type Package

Title Spatial Downscaling using the Dissever Algorithm

Version 0.2-3

Date 2018-04-20


URL https://github.com/pierreroudier/dissever

BugReports https://github.com/pierreroudier/dissever/issues

License GPL-2

LazyData TRUE

Imports  methods, caret, sp, raster, plyr, dplyr, foreach, boot, magrittr, viridis

Suggests knitr, rmarkdown, randomForest, Cubist, gam, stats

Collate 'dissiever-package.R' 'dissiever.R'

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Author Pierre Roudier [cre, aut], Brendan Malone [ctb]

Maintainer Pierre Roudier <roudierp@landcareresearch.co.nz>

Repository CRAN

Date/Publication 2018-04-20 04:03:06 UTC
\textbf{R topics documented:}

- dissever-package .................................................. 2
- dissever ............................................................. 2
- edgeroi ............................................................... 4
- generate_ci .......................................................... 6
- plot.dissever .......................................................... 7
- print.dissever .......................................................... 7
- summary.dissever ..................................................... 8

\textbf{Description}

Dissever: algorithm for spatial downscaling

\textbf{Details}

Algorithm described in: [DOI: 10.1016/j.cageo.2011.08.021]

\textbf{Author(s)}

Brendan Malone (brendan.malone@sydney.edu.au);

\textbf{Description}

Performs spatial downscaling of coarse grid mapping to fine grid mapping using predictive covariates and a model fitted using the caret package.

\textbf{Usage}

```r
## S4 method for signature 'RasterLayer,RasterLayer'
dissever(coarse, fine, method = "rf",
P = 0.5, mmax = NULL, thresh = 0.01, min_iter = 5, max_iter = 20,
boot = NULL, level = 0.9, tune_length = 3,
tune_grid = .create_tune_grid(model = method, tune_length = tune_length),
train_control_init = .default_control_init,
train_control_iter = .default_control_iter, verbose = FALSE)
```
**Arguments**

- **coarse** object of class "RasterLayer", the coarse-resolution layer that needs to be downscaled.
- **fine** object of class "RasterStack", the fine-resolution stack of predictive covariates.
- **method** a string specifying which classification or regression model to use (via the caret package). Possible values are found using names(caret::getModelInfo()).
- **p** numeric, proportion of the fine map that is sampled for fitting the dissever model (between 0 and 1, defaults to 0.5)
- **nmax** numeric maximum number of pixels selected for fitting the dissever model. It will override the number of pixels chosen by the p option if that number is over the value passed to nmax.
- **thresh** numeric, dissever iterations will proceed until the RMSE of the dissever model reaches this value, or until the maximum number of iterations is met (defaults to 0.01)
- **min_iter** numeric, minimum number of iterations (defaults to 5)
- **max_iter** numeric, maximum number of iterations (defaults to 20)
- **boot** numeric, if not NULL (default), the number of bootstrap replicates used to derive the confidence intervals.
- **level** If this is a numeric value, it is used to derive confidence intervals using quantiles. If it is a function, it is used to derive the uncertainty using this function.
- **tune_length** numeric, the number of parameters to test to find the optimal parametrisation of the caret model (defaults to 3)
- **tune_grid** a data frame with possible tuning values
- **train_control_init** Control parameters for finding the optimal parameters of the caret model (see trainControl)
- **train_control_iter** Control parameters for fitting the caret model during the iteration phase (see trainControl)
- **verbose** controls the verbosity of the output (TRUE or FALSE)

**Author(s)**

Brendan Malone, Pierre Roudier

**References**

Examples

# Load the Edgeroi dataset (see ?edgeroi)
data(edgeroi)

# Plot the Edgeroi dataset (using the raster package)
library(raster)
plot(edgeroi$carbon) # coarse resolution layer
plot(edgeroi$predictors) # fine resolution predictors

# Run dissever using a simple linear model.
# In this instance we are subsampling heavily (p = 0.05) to keep
# run time short
res_lm <- dissever(
  coarse = edgeroi$carbon,
  fine = edgeroi$predictors,
  method = "lm",
  min_iter = 5, max_iter = 10,
  p = 0.05
)

# A lot more models are available through caret:
## Not run:
subset(caret::modelLookup(), forReg == TRUE, select = 'model')
## End(Not run)

# Plot dissever results
plot(res_lm, type = 'map', main = "Dissever using GAM")
plot(res_lm, type = 'perf', main = "Dissever using GAM")

---

edgeroi

Subset of the 1-km resolution soil carbon map and selected subset of environmental covariates for the Edgeroi District, NSW

Description

Subset of the Australian 1-km resolution soil carbon map, along with selected environmental covariates, for a small area of the Edgeroi district in NSW, Australia.

Usage

data(edgeroi)

Format

The edgeroi dataset is a list with two elements:
carbon a 7 row, 10 column RasterLayer of soil carbon stocks for a small area in the Edgeroi district NSW, Australia. The grid has a pixel resolution of 1 x 1 km. It contains the predicted soil carbon stock for the 0-30cm depth

predictors an 87 row, 117 column, 5 layer RasterStack of selected environmental covariates from a small area in the Edgeroi district NSW, Australia. The grids have a pixel resolution of 90m x 90m. It contains the following layers:

Doserate numeric; gamma radiometric data
Elevation numeric; topographic variable of bare earth ground elevation. Derived from digital elevation model
Panchromat numeric; panchromatic band of the Landsat 7 satellite
Slope numeric; Slope gradient of the land surface. Derived from digital elevation model
TWI numeric; topographic wetness index. Secondary derivative of the digital elevation model

Details

The Edgeroi District, NSW is an intensive cropping area upon the fertile alluvial Namoi River plain. The District has been the subject of many soil investigations, namely McGarry et al. (1989) whom describe an extensive soil data set collected from the area. More recently, digital soil mapping studies of the area have been conducted, for example, Malone et al. (2009). The 1km mapping of soil carbon stock was performed using digital soil mapping methods McBratney et al. (2003) using the soil data from McGarry et al. (1989) and available environmental covariates.

Note

The raw spatial data that contributed to the creation of edgeGrids were sourced from publically accessable repositories hosted by various Australian Government and international agencies including CSIRO (for the DEM), Geosciences Australia (for the radiometric data) and NASA (for the Landsat 7 ETM+ data). The projection for the RasterStack is WGS 84 Zone 55.

References

generate_ci

Confidence intervals using bootstrapping

Description

Generates confidence intervals of a dissever output using bootstrapping.

Usage

```r
## S4 method for signature 'list,RasterStack'
generate_ci(object, covariates, level = 0.9,
             n = 50L)
```

Arguments

- **object**: object of class `dissever`, output from the `dissever` function.
- **covariates**: object of class "RasterStack", the fine-resolution stack of predictive covariates used to generate the dissever output.
- **level**: If this is a numeric value, it is used to derive confidence intervals using quantiles. If it is a function, it is used to derive the uncertainty using this function.
- **n**: the number of bootstrap replicates used to derive the confidence intervals.

Author(s)

Pierre Roudier

Examples

```r
# Load the Edgeroi dataset (see ?edgeroi)
data(edgeroi)

# Create a dissever output
diss <- dissever(
  coarse = edgeroi$carbon,
  fine = edgeroi$predictors,
  method = "lm",
  min_iter = 5, max_iter = 10,
  p = 0.05
)

# Generate the confidence intervals
## Not run:
ci <- generate_ci(diss, edgeroi$predictors, n = 5)
plot(ci)

## End(Not run)
```
**plot.dissever**

*Plots a dissever result*

---

**Description**

Plots a dissever result. Two modes are possible to visualise either the resulting map or the convergence of the disseveration.

**Usage**

```r
## S3 method for class 'dissever'
plot(x, type = "map", ...)
```

**Arguments**

- `x`: object of class dissever, output from the dissever function
- `type`: character, type of visualisation to produce. Options are "map", to produce a map of the dissevered coarse map, or "perf", to show the convergence of the RMSE during the disseveration procedure.
- `...`: Additional arguments passed to `plot`

**Author(s)**

Pierre Roudier

**Examples**

```
# See ?dissever
```

---

**print.dissever**

*Prints the performance of the dissever procedure*

---

**Description**

Prints the performance of the model used to do the dissever procedure.

**Usage**

```r
## S3 method for class 'dissever'
print(x, ...)
```

**Arguments**

- `x`: object of class dissever, output from the dissever function
- `...`: Additional arguments passed to `print`
**Author(s)**

Pierre Roudier

---

**summary.dissever**

*Prints summary of the model used in the dissever procedure*

---

**Description**

Prints summary of the model used in the dissever procedure.

**Usage**

```r
## S3 method for class 'dissever'
summary(object, ...)
```

**Arguments**

- `object` object of class `dissever`, output from the `dissever` function
- `...` Additional arguments passed to `summary`

**Author(s)**

Pierre Roudier
Index

dissever, 2

dissever,RasterLayer,RasterLayer-method
  (dissever), 2

dissever,RasterLayer,RasterStack-method
  (dissever), 2

dissever-package, 2

edgeroi, 4

generate_ci, 6

generate_ci,list,RasterStack-method
  (generate_ci), 6

plot.dissever, 7

print.dissever, 7

summary.dissever, 8