

Package ‘clhs’

October 14, 2016

Type Package

Title Conditioned Latin Hypercube Sampling

Version 0.5-6

Date 2016-10-10

Author Pierre Roudier

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

Description Conditioned Latin hypercube sampling, as published by Minasny and McBratney (2006) <DOI:10.1016/j.cageo.2005.12.009>. This method proposes to stratify sampling in presence of ancillary data. An extension of this method, which propose to associate a cost to each individual and take it into account during the optimisation process, is also proposed (Roudier et al., 2012, <DOI:10.1201/b12728-46>).

Depends R (>= 2.14.0)

Imports utils, methods, grid, ggplot2, sp, raster, reshape2, plyr, scales

License GPL (>= 2)

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2016-10-14 00:10:02

R topics documented:

clhs-package	2
clhs	3
cLHS_result	5
plot.cLHS_result	5

Index	7
--------------	----------

Description

This package implements the conditioned Latin hypercube sampling, as published by Minasny and McBratney (2006). This method proposes to stratify sampling in presence of ancillary data.

An extension of this method, which propose to associate a cost to each individual and take it into account during the optimisation process, is also proposed (Roudier et al., 2012).

Author(s)

Pierre Roudier

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

References

*For the initial cLHS method:

Minasny, B. and McBratney, A.B. 2006. A conditioned Latin hypercube method for sampling in the presence of ancillary information. *Computers and Geosciences*, 32:1378-1388.

*For the cost-constrained implementation:

Roudier, P., Beaudette, D.E. and Hewitt, A.E. 2012. A conditioned Latin hypercube sampling algorithm incorporating operational constraints. In: *Digital Soil Assessments and Beyond. Proceedings of the 5th Global Workshop on Digital Soil Mapping*, Sydney, Australia.

See Also

sample

Examples

```
df <- data.frame(
  a = runif(1000),
  b = rnorm(1000),
  c = sample(LETTERS[1:5], size = 1000, replace = TRUE)
)
res <- clhs(df, size = 50, iter = 2000, progress = FALSE)
str(res)
```

`clhs`*Conditioned Latin Hypercube Sampling*

Description

Implementation of the conditioned Latin hypercube sampling, as published by Minasny and McBratney (2006). This method proposes to stratify sampling in presence of ancillary data. An extension of this method, which propose to associate a cost to each individual and take it into account during the optimisation process, is also proposed (Roudier et al., 2012).

Usage

```
clhs(x, ...)
```

Arguments

<code>x</code>	A <code>data.frame</code> , <code>SpatialPointsDataFrame</code> or <code>Raster</code> object.
<code>...</code>	Additional arguments, see under <code>Details</code> .

Details

Below are the additional arguments that can be used:

- `size`: A non-negative integer giving the number of samples to pick.
- `cost`: A character giving the name or an integer giving the index of the attribute in `x` that gives a cost that can be use to constrain the cLHS sampling. If `NULL` (default), the cost-constrained implementation is not used.
- `track`: A character giving the name or an integer giving the index of the attribute in `x` that gives a cost associated with each individual. However, this method will only track the cost - the sampling process will not be constrained by this attribute. If `NULL` (default), this option is not used.
- `iter`: A positive number, giving the number of iterations for the Metropolis-Hastings annealing process. Defaults to 10000.
- `temp`: The initial temperature at which the simulated annealing begins. Defaults to 1.
- `tdecrease`: A number between 0 and 1, giving the rate at which temperature decreases in the simulated annealing process. Defaults to 0.95.
- `weights`: A list a length 3, giving the relative weights for continuous data, categorical data, and correlation between variables. Defaults to `list(numeric = 1, factor = 1, correlation = 1)`.
- `obj.limit`: The minimal value at which the optimisation is stopped. Defaults to `-Inf`.
- `length.cycle`: The duration (number of iterations) of the isotherm steps. Defaults to 10.
- `progress`: `TRUE` or `FALSE`, displays a progress bar.
- `simple`: `TRUE` or `FALSE`. If set to `TRUE`, only the indices of the selected samples are returned, as a numeric vector. If set to `FALSE`, a `cLHS_result` object is returned (takes more memory but allows to make use of `cLHS_results` methods such as `plot.cLHS_result`).

Value

* If the `simple` option is set to `TRUE` (default behaviour): A numeric vector containing the indices of the selected samples is returned

* If the `simple` option is set to `FALSE`: An object of class `cLHS_result`, with the following elements:

<code>index_samples</code>	a vector giving the indices of the chosen samples.
<code>sampled_data</code>	the sampled data.frame.
<code>obj</code>	a vector giving the evolution of the objective function throughout the Meropolis-Hastings iterations.
<code>cost</code>	a vector giving the evolution of the cost function throughout the Meropolis-Hastings iterations (if available).

Author(s)

Pierre Roudier

References

*For the initial cLHS method:

Minasny, B. and McBratney, A.B. 2006. A conditioned Latin hypercube method for sampling in the presence of ancillary information. *Computers and Geosciences*, 32:1378-1388.

*For the cost-constrained implementation:

Roudier, P., Beaudette, D.E. and Hewitt, A.E. 2012. A conditioned Latin hypercube sampling algorithm incorporating operational constraints. In: *Digital Soil Assessments and Beyond*. Proceedings of the 5th Global Workshop on Digital Soil Mapping, Sydney, Australia.

See Also

[plot.cLHS_result](#)

Examples

```
df <- data.frame(
  a = runif(1000),
  b = rnorm(1000),
  c = sample(LETTERS[1:5], size = 1000, replace = TRUE)
)

# Returning the indices of the sampled points
res <- clhs(df, size = 50, iter = 100, progress = FALSE, simple = TRUE)
str(res)

# Returning a cLHS_result object for plotting
res <- clhs(df, size = 50, iter = 100, progress = FALSE, simple = FALSE)
str(res)
plot(res)
```

cLHS_result	<i>Conditioned Latin Hypercube Sampling result</i>
-------------	--

Description

A S3 class describing a cLHS result.

Value

An object of class `cLHS_result` contains the following slots:

<code>index_samples</code>	a vector giving the indices of the chosen samples.
<code>sampled_data</code>	the sampled data.frame.
<code>obj</code>	a vector giving the evolution of the objective function throughout the Metropolis-Hastings iterations.
<code>cost</code>	a vector giving the evolution of the cost function throughout the Metropolis-Hastings iterations, if available, otherwise NULL.

Author(s)

Pierre Roudier

See Also

`clhs`

<code>plot.cLHS_result</code>	<i>Plot cLHS results</i>
-------------------------------	--------------------------

Description

Produces a plot illustrating the result of a cLHS sampling procedure.

Usage

```
## S3 method for class 'cLHS_result'
plot(x, modes = "obj", ...)
```

Arguments

<code>x</code>	Object of class “cLHS_result”.
<code>modes</code>	A character vector describing the plot to produce (see Details)
<code>...</code>	Other ggplot2 plotting parameters.

Details

The subplots to be included in the final illustration are controlled by the `mdoe` option: - "obj" adds the evolution of the objective function over the iterations - "cost" adds the evolution of the cost function over the iterations (if available in `x`) - "hist" adds the comparison of the distributions of each variables in both the original object and the sampled result using histogram plots (for continuous variables). - "dens" adds the comparison of the distributions of each variables in both the original object and the sampled result using density plots (for continuous variables). - "box" adds the comparison of the distributions of each variables in both the original object and the sampled result using boxplots (for continuous variables).

Author(s)

Pierre Roudier

See Also

[clhs](#)

Examples

```
df <- data.frame(
  a = runif(1000),
  b = rnorm(1000),
  c = sample(LETTERS[1:5], size = 1000, replace = TRUE)
)
res <- clhs(df, size = 50, iter = 2000, progress = FALSE, simple = FALSE)
# You can plot only the objective function
plot(res, mode = "obj")
# Or you can compare the distribution in the original object
# and in the sampled result
plot(res, mode = c("obj", "box"))
```

Index

*Topic **package**

clhs-package, 2

clhs, 3, 6

clhs, data.frame-method (clhs), 3

clhs, Raster-method (clhs), 3

clhs, SpatialPointsDataFrame-method
(clhs), 3

clhs-package, 2

clhs.data.frame (clhs), 3

clhs.Raster (clhs), 3

clhs.SpatialPointsDataFrame (clhs), 3

cLHS_result, 5

plot.cLHS_result, 4, 5