Package 'hdeco'

April 17, 2009

Type Package

Title Hierarchical DECOmposition of Entropy for Categorical Map Comparisons

Version 0.4.1

Date 2009-02-25

Author Tarmo K. Remmel, Sandor Kabos, and Ferenc (Ferko) Csillag

Maintainer Tarmo K. Remmel <remmelt@yorku.ca>

Description A flexible and hierarchical framework for comparing categorical map composition and configuration (spatial pattern) along spatial, thematic, or external grouping variables. Comparisons are based on measures of mutual information between thematic classes (colours) and location (spatial partitioning). Results are returned in textual, tabular, and graphical forms.

Encoding latin1

License GPL (>= 2)

Depends R(>= 2.0.1)

Lazydata yes

Repository CRAN

Date/Publication 2009-02-25 08:29:20

R topics documented:

`hdeco-package` *Hierarchical DECOmposition of Entropy for Categorical Map Comparisons*

Description

We provide a measurement scheme for the comparison of categorical maps that decomposes the differences in multidimensional nested coincidence tables according to variables that record occurrence frequencies of categories (Z), at levels of spatial aggregation (Y), on specific maps (X). Sequences of conditional entropies computed according to the specific questions asked (e.g. is there coincidence between colours and locations), characterize the correspondence between the three types of variables in common units (bits) measured by mutual information. The form of these sequences, as a variable runs from coarse to fine detail, referred to as spectra, provide meaningful characterizations of the similarities/differences between categorical maps, including their spatial configuration.

Details

Package:	hdeco
Type:	Package
Version:	0.4
Date:	2007-05-02
License:	GPL (Version 2.0 or later)

Author(s)

Tarmo K. Remmel and Sandor Kabos
Maintainer: Tarmo K. Remmel <remmelt@yorku.ca>

References

Remmel, T.K. and F. Csillag. 2006. Mutual information spectra for comparing categorical maps. International Journal of Remote Sensing 27(7)1425-1452.

Remmel, T.K., F. Csillag, and S. Kabos. 2004. Using hierarchical entropy decomposition to compare the spatial patterns found in categorical digital maps. In Abstracts of the Canadian Association of Geographers (CAG) Annual Meeting, May 25-29, Moncton, New Brunswick, Canada.

`CARsimu` *Conditional AutoRegressive surface simulator*

Description

A stationary tool for simulating a surface (as a matrix of values) with specified spatial autocorrelation parameters. The simulation return $2^level * 2^level$ real-valued surface (e.g., landscape) based on the FFT algorithm and the spectral (or autocorrelation theorem using first and second order neighbours in N-S, E-W, NW-SE, and NE-SW directions.

Usage

```
CARsimu(level = 5, row1 = 0.2499999, row2 = 0, col1 = 0.2499999, col2 = 0, rc1 = 0,
```

Arguments

`level`	Integer: controls the dimensions of the output surface ($2^level * 2^level$)
`row1`	First order neighbour East-West autocorrleation parameter
`row2`	Second order neighbour East-West autocorrelation parameter
`col1`	First order neighbour North-South autocorrelation parameter
`col2`	Second order neighbour North-South autocorrelation parameter
`rc1`	First order neighbour NW-SE autocorrelation parameter
`cr1`	First order neighbour NE-SW autocorrelation parameter

Details

The sum of all six autocorrelation parameters MUST be less than 0.5. Their effect is highly nonlinear, thus there is a marked difference beetween 0.499 and 0.499999. This implementation corresponds to the homogeneous (stationary) conditional autoregressive (CAR) model.

Value

Returns a $2^level * 2^level$ matrix of real numbers. It is possible to use the image() or imaks() functions to graphically display the surface. Saving the output from this function into a new object is likely the most desireable usage. Ths surface can then be factored to produce binary maps with specified proportions as in Remmel and Csillag (2003).

Note

This function was originally called ujabki() - some earlier versions may be available elsewhere under this name.

Author(s)

Sandor Kabos, Ferenc (Ferko) Csillag, and Tarmo Remmel

References

Remmel, T.K. and F. Csillag. 2003. When are two landscape pattern indices significantly different? Journal of Geographical Systems 5(4):331-351.

Examples

```
carimage <- CARsimu(level=5, row1=0.2499999, row2=0, col1=0.2499999, col2=0, rc1=0, cr1=0)
image(carimage)
```

`checker` *Checkerboard map builder*

Description

A simple function for building 64x64 pixel checkerboard (categorical binary) maps with specified block sizes to be used as test maps.

Usage

```
checker(block = 4)
```

Arguments

`block`	The number of pixels that form the edge of a single colour block. For example, 4 will yield a checkerboard with 16 pixel squares.

Value

This function returns a binary integer matrix with dimensions 64x64.

Note

This function only generates 64x64 binary maps for testing. Construction of larger maps will require a modificaiton of the source code.

Author(s)

Tarmo K. Remmel

Examples

```
checker(block = 4)
```

decomppath *Demonstration Decomposition Path Matrix*

Description

A decomposition path matrix for use with the supplied demonstration data. It is inteded for use for comparing two 16x16 pixel categorical maps along a spectra of spatial decomposition.

Usage

```
data(decomppath)
```

Examples

```
data(decomppath)
str(decomppath)
decomppath
```

demoimage1 *Demonstration binary 16x16 categorical map 1*

Description

For use with hdeco (and other functions) to illustrate usage. This categorical map has two vertical bars.

Usage

```
data(demoimage1)
```

Examples

```
data(demoimage1)
str(demoimage1)
fixedcolimage(demoimage1)
```

demoimage2 *Demonstration binary 16x16 categorical map 2*

Description

For use with hdeco to illustrate usage. This categorical map has 2 horizontal bars.

Usage

```
data(demoimage1)
```

Examples

```
data(demoimage2)
str(demoimage2)
fixedcolimage(demoimage2)
```

fixedcolimage *Categorical map drawing tool with set colour scheme*

Description

A tool for drawing categorical data (matrices) in their proper orientation and with a set colour palette. This is especially useful for drawing multiple images that requrie identical palettes even though some classes may not exist on all images.

Usage

```
fixedcolimage(BE = checker(8))
```

Arguments

BE	Matrix: this is the input image that may or may not have an attribute cim that contains the title text.

Details

The palette is currently fixed for 21 colours. If more categorical colours are desired, the source code will need to be modified by the addition of more colours.

Value

The result is a graphic with specified colour palette. The original palette along with the used colours are listed in the command window for reference.

Note

The original function, imaks, did not force a common colour scheme across multiple images. This version assigns specific colours to integer values and maintains class–colour consistency across multiple categorical images. Images can be assigned titles with an attribute called cim that will be placed at the top of the image during plotting.

Author(s)

Tarmo K. Remmel, modified from Sandor Kabos

Examples

```
test <- checker(8)
attr(test, "cim") <- "Checker 8"
fixedcolimage(test)
```

---

hdeco *Hierarchical DECOmposition Analysis Environment*

Description

A flexible data model and analysis environment for comparing categorical maps. Using information theory, differences between (or among) maps are computed along trajectories as either space or colours are decomposed from coarse to fine aggregations.

Usage

```
hdeco(BE1 = demoimage1, BE2 = demoimage2, MICIKE = decomppath, MSK = FALSE, MASK =
```

Arguments

BE1	Required: Input categorical map 1 matrix object.
BE2	Input categorical map 2 matrix object (Required if comparing BE1 to BE2).
MICIKE	Required: The decomposition path definition matrix.
MSK	Boolean: TRUE if automatic filtering of zero values is to take place, otherwise FALSE
MASK	If a mask is to be applied, enter the object name.
fnev	ASCII filename to where text results are to be written. If NULL, results are displayed in the command window.
AutoDecoPath	Boolean: if TRUE, a default decomposition path is used - use extreme caution, this is not recommended!
JPG	Boolean: TRUE if graphic results are to be saved in JPEG format, otherwise FALSE to have graphics only displayed in a graphics window.
zsir	Outdated and should be removed in a subsequent version - ignore.

MULTIX	Boolean: TRUE if multiple X-variables will be used, otherwise FALSE.
RECODEX	Boolean: TRUE if X-variables will be recoded.
NXRECODES	Integer: Number of X-variables recodes will be provided by the lookup table.
LUTX	Lookup table for recoding X-variables.
RECODEZ	Boolean: TRUE if Z-variables will be recoded, otherwise FALSE.
NZRECODES	Integer: Indicate how many Z-variable recodes will be provided by the lookup table.
LUTZ	Lookup table for recoding Z-variables.
HISTOGRAM	Boolean: TRUE if histograms of the map(s) are to be drawn, otherwise FALSE.
Z1DROP	Boolean: TRUE if the Z1-variable is to be dropped from the multidimensional array construction. If the Z-variable is recoded and the original data (Z1) are not required, dropping this variable can save considerable space and processing time.
OMITX1	Boolean: TRUE if the X1-variable is to be dropped from the multidimensional array construction. If the Z-variable is recoded and the original data (Z1) are not required, dropping this variable can save considerable space and processing time.
PS	Bolean: TRUE if Postscript versions of the graphics are to be generated, otherwise FALSE.

Details

Graphical results are returned to the graphics window, a series of tabular results are stored as hidden objects. To view a listing of these hidden objects, use the provided function ls.().

Value

.N	The cardinality of map and spatial decompositions.
.QND	The cardinality vector for the specified decomposition.
.QKEP	The multi-dimensional array of probabilities.
.VFONAL	The decomposition pathway matrix used in the latest run of HDECO.
.MASKTITLE	The name of the mask used, if specified by the cim attribute.
.CIM	The name of the image(s) processed, if specified by the cim attribute(s).
.COLOURS	The total number of colours entering the HDECO algorithm.
.LUT.X	The lookup table for the X variable(s) if used; otherwise NULL.
.LUT.Z	The lookup table for the Z variable(s) if used; otherwise NULL.
.BASE	The base hypotheses - all X, Y, and Z variables entering the analysis.
.HPROFIL	The primary output of hdeco used for producing graphical spectra. The columns represent the joint entropy between null and alternate hypotheses (HALAPF), the entropy of the null hypothesis (HNULL), the entropy of the alternate hypothesis (HALT), the mutual information between the null and alternate hypotheses (MUTU), and the uncertainty coefficient (UNC). Additionally, the G-squared statistic, it's significance value, an indication (SING-MULT=1,2) as to whether

	the decomposition was for a single or multiple images, and a flag that determines log file phrase selection (DESC=1,2) reflecting either single or multiple image decomposition.
.AHIPO	The alternate hypotheses at each decomposition step - the X, Y, and Z variables comprising the alternate hypotheses.
.NHIPO	The null hypotheses at each decomposition step - the X, Y, and Z variables comprising the null hypotheses.
.KIVALO	The conditional alternate hypotheses identified at each step of the decomposition. These are identified by the integer '2' in the decomposition pathway matrix.

Note

If 2 or more images are entered into the decomposition, they must have identical dimensions. Furthermore, the dimensions must adhere to the $2^N * 2^N$ constraint. The t.forcesize() function can be used to inset a smaller image into a larger nodata image that can later be masked out for processing purposes. If the fnev argument is provided, the detailed output that is normally written to the screen is dumped into a log file with the basename provided in the fnev argument string.

Author(s)

Tarmo K. Remmel and Sándor Kabos

Examples

```
data(demoimage1)
data(demoimage2)
data(decomppath)
hdeco(BE1=demoimage1, BE2=demoimage2, MICIKE=decomppath)
```

HPROFIL *Sample HDECO output data*

Description

A sample .HPROFIL matrix renamed to HPROFIL that can be used to test the plotting functions of sigplot etc.

Usage

```
data(HPROFIL)
```

See Also

See Also sigplot, hdeco

Examples

```
data(HPROFIL)
str(HPROFIL)
HPROFIL
```

---

imaks *Categorical map drawing tool with set colour scheme*

Description

A tool for drawing categorical data (matrices) in their proper orientation and with a set colour palette. This is especially useful for drawing multiple images that requrie identical palettes even though some classes may not exist on all images.

Usage

```
imaks(BE = demoimage1, ncolours=NULL, LENG=4)
```

Arguments

BE	Matrix: this is the input image that may or may not have an attribute cim that contains the title text.
ncolours	Integer: the total number of colours in the image. This can be automatically determined if set to NULL.
LENG	integer: used to control labelling.

Value

The result is a graphic with title read from attribute 'cim' if it exists.specified colour palette. This function is suitable for use with continuous surfaces.

Author(s)

Sandor Kabos and Tarmo Remmel

Examples

```
data(demoimage1)
imaks(demoimage1)
```

`ls.` *List objects starting with '.'*

Description

Similar to the function ls, this modified form shows only those objects whose names begin with a period.

Usage

```
ls.()
```

Details

This function will only display the hidden objects in a workspace. To see other objects, use the traditional form of the list function, ls

Value

Provides a listing of all objects in the workspace whose names begin with a period.

Note

Useful for listing hidden objects in a workspace that are generated by HDECO

Author(s)

Ferenc Csillag

See Also

`hdeco`, `ls`

Examples

```
ls.()
```

`makepath` *Tool for building decomposition pathways for HDECO*

Description

A tool for specifying the dimensions of a new decomposition pathway that provides a spreadsheet-style entry and editing environment for cell values.

Usage

```
makepath(NX = 2, NY = 6, NZ = 2, STEPS = 7, TITLE = "Custom Decomposition Path")
```

Arguments

`NX`	Integer: How many X-variables to add as columns in the decomposition path
`NY`	Integer: How many Y-variables to add as columns in the decomposition path
`NZ`	Integer: How many Z-variables to add as columns in the decomposition path
`STEPS`	Integer: How many rows, or decomposition steps, are to appear in the decomposition path
`TITLE`	String: A custom name given to this decomposition path

Details

The decomposition path dimensions must match the data entering the comparison. Each line will be executed sucessively and will control the operation of HDECO. Proper definition of the decomposition path is elemental to the successful running of HDECO. Once the dimensions are set, this function will provide a spreadsheet-style entry envirionment where the cell entries must be provided. When the spreadsheet is closed, the results are saved to an object called VFONAL. Subsequent calls to makepath will overwrite VFONAL; to mitigate this effect, rename VFONAL after its creation.

Value

This function returns a matrix with the dimensions defined above; it is called VFONAL. It will have an attribute cim that stores the custom title given to this decomposition path.

Note

Each row of the decomposition path is executed independently. Valid definition of the decomposition path is essential to the operation of HDECO. Numerous meaningless decomposition paths are possible. If unsure, begin with the examples provided in the references.

Author(s)

Tarmo K. Remmel

References

Remmel, T.K. and F. Csillag. 2006. Mutual information spectra for comparing categorical maps. International Journal of Remote Sensing 27(7)1425-1452.

See Also

hdeco

`shift` *A tool for shifting a categorical map horizontally or vertically on a torus*

Description

Use this function to shift a categorical map by a specified number of pixels up, down, left, or right. The shifted pixels are wrapped around to the opposite edge of the map.

Usage

```
shift(map = demoimage1, dir = 1, n = 1, draw = TRUE, verbose = TRUE)
```

Arguments

`map`	A required input categorical map as a matrix. Currently, the requirement is that the x and y dimensions are equal.
`dir`	An integer (1,2,3, or 4) that identifies the shift direction: up, right, down, or left respectively.
`n`	The number of pixels to shift.
`draw`	A boolean flag indicating whether results will be drawn to the graphics window.
`verbose`	A boolean flag indicating whether verbose feedback is provided to standard output.

Value

If valid parameters are set, the output will be the shifted map object as a matrix. Otherwise, error messages are provided and NULL is returned.

Note

This results of this function should be saved to an object.

Author(s)

Tarmo K. Remmel

See Also

See Also fixedcolimage, demoimage1

Examples

```
data(demoimage1)
shift(demoimage1, dir=2, n=3, draw=TRUE, verbose=TRUE)
```

`sigplot` *A HDECO internal plot tool for identifying G-squared significances on spectra.*

Description

Adds shaded points to decomposition spectra to identify decomposition steps with significant G-squared2 statistic values. Requires the output from t.hdeco, specifically the object titled .HPROFIL. This function is called internally by t.hdeco, but can be used independently if the .HPROFIL object exists.

Usage

```
sigplot(mat = .HPROFIL, column = 5, sigcol = 7, tit = "Title of Graph", xtit = "Ste
```

Arguments

`mat`	The required .HPROFIL matrix output by the t.hdeco function (or a comparable matrix object).
`column`	Identifies the column (integer) with the data to be plotted (e.g., 5 is the uncertainty coefficient - UNC).
`sigcol`	Identifies the column (integer) with the significance value for the G-squared statistic.
`tit`	A text string identifying the title for the plot.
`xtit`	A text string used for the x-axis label.
`ytit`	A text string used for the y-axis label.
`override`	Boolean flag to override the minimum and maximum y-axis values.
`lowy`	If override is TRUE, then the new low y-axis value.
`highy`	If override is TRUE, then the new high y-axis value.

Value

The graphical result is provided in the graphics window.

Author(s)

Tarmo K. Remmel

Examples

```
data(HPROFIL)
sigplot(mat = HPROFIL, column = 5, sigcol = 7, tit = "Title of Graph", xtit = "Step", ytit =
```

Index