Package 'integr'

May 8, 2026

Type Package

Title An Implementation of Interaction Graphs of Aleks Jakulin

Version 1.0.0

Description Generates a 'Graphviz' graph of the most significant 3-way interaction gains (i.e. conditional information gains) based on a provided discrete data frame. Various output formats are supported ('Graphviz', SVG, PNG, PDF, PS). For references, see the webpage of Aleks Jakulin <http://stat.columbia.edu/~jakulin/Int/>.

Depends R (>= 3.5.0), dplyr (>= 0.7.6), DiagrammeR (>= 1.0.0), DiagrammeRsvg (>= 0.1), rsvg (>= 1.3), gtools (>= 3.5.0), utils (>= 3.5.0)

License GPL-3

Encoding UTF-8

LazyData true

URL https://github.com/peleplay/integr

BugReports https://github.com/peleplay/integr/issues

RoxygenNote 6.1.1

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Author Petar Markovic [aut, cre]

Maintainer Petar Markovic <peleplay@gmail.com>

Repository CRAN

Date/Publication 2019-05-24 11:50:03 UTC

Contents

`entropy` *Calculates Shannon's entropy*

Description

Formula: $H(S) = -P_i * \sum log_2 * P_i$, where P_i is the probability of the corresponding i-th class

Usage

```
entropy(df, classAtt)
```

Arguments

`df`	A discrete `data.frame`
`classAtt`	A class column of the df (`string`)

Value

The Shannon's entropy of the `df`, based on the `classAtt` attribute

Examples

```
entropy(golf, "Play")
```

golf *Golf example dataset for Interaction graphs*

Description

An example dataset containing the discrete data.frame (i.e. all columns are factors) with variables used as an input for making a decision whether a party of golf would be played, or not.

Usage

```
golf
```

Format

A data.frame with 6 discrete variables (i.e. factors) and 14 rows (i.e. observations). 5 input variables and 1 class (i.e. context) variable:

Outlook Input attribute, values: Overcast, Rainy, Sunny

Temperature Input attribute, values: Cool, Hot, Mild

Humidity Input attribute, values: High, Normal

Windy Input attribute, values: True, False

Others Artifically added input attribute indicating whether the players on the other courts were playing the golf at the given time, values: Yes, No

Play Class attribute, indicating whether the decision was to play or not to play a party of golf, values: Yes, No

@source https://gerardnico.com/data_mining/weather

ig *Constructs Interaction Graph (S3 class)*

Description

Constructs Interaction Graph (S3 class)

Usage

```
ig(n, e)
```

Arguments

n	ig.nodes (a list of igNode objects)
e	ig.edges (a list of igEdge objects)

Value

An instance of the ig class

igEdge *Constructs Interaction Graph Edges (S3 class)*

Description

Constructs Interaction Graph Edges (S3 class)

Usage

```
igEdge(n1, n2, w)
```

Arguments

n1	igEdge.node1 (character)
n2	igEdge.node2 (character)
w	igEdge.weight (i.e. 3-way Interaction Gain) (double)

Value

An instance of the igEdge class

igNode *Constructs Interaction Graph Nodes (S3 class)*

Description

Constructs Interaction Graph Nodes (S3 class)

Usage

```
igNode(n, v)
```

Arguments

n	igNode.name (character)
v	igNode.value (double) (i.e. 2-way Interaction Gain)

Value

An instance of the igNode class

igToGrViz *Exports Interaction graph to a GraphViz file*

Description

Exports Interaction graph to a GraphViz file

Usage

```
igToGrViz(ig, path = "", fName = "InteractionGraph")
```

Arguments

ig	Interaction graph
path	The folder in which to write the GraphViz file;
fName	The name of the file to be created; "InteractionGraph" by default

Value

Writes the ig interaction graph to a GraphViz .gv file to the folder specified in the path

Examples

```
#create temp dir path with slashes
myDir <- gsub("\\\\", "/", tempdir())

#create interaction graph
g <- interactionGraph(golf, "Play", intNo = 10)

#write to 'graphviz' file
igToGrViz(g, path = myDir, fName = "MyGraph")
```

igToPDF *Exports Interaction graph to a PDF file*

Description

Exports Interaction graph to a PDF file

Usage

```
igToPDF(ig, path = "", fName = "InteractionGraph", h = 2000)
```

Arguments

ig	Interaction graph
path	The folder in which to write the PDF file;
fName	The name of the file to be created; "InteractionGraph" by default
h	Desired height of the image in pixels; 2000px by default

Value

Writes the ig interaction graph to a PDF (.pdf) file to the folder specified in the path

Examples

```
#create temp dir path with slashes
myDir <- gsub("\\\\", "/", tempdir())

#create interaction graph
g <- interactionGraph(golf, "Play", intNo = 10)

#write to PDF
igToPDF(g, path = myDir, fName = "MyGraph", h = 2000)
```

---

igToPNG *Exports Interaction graph to a PNG file*

Description

Exports Interaction graph to a PNG file

Usage

```
igToPNG(ig, path = "", fName = "InteractionGraph", h = 2000)
```

Arguments

ig	Interaction graph
path	The folder in which to write the PNG file;
fName	The name of the file to be created; "InteractionGraph" by default
h	Desired height of the image in pixels; 2000px by default

Value

Writes the ig interaction graph to a PNG (.png) file to the folder specified in the path

Examples

```
#create temp dir path with slashes
myDir <- gsub("\\\\", "/", tempdir())

#create interaction graph
g <- interactionGraph(golf, "Play", intNo = 10)

#write to PNG
igToPNG(g, path = myDir, fName = "MyGraph", h = 2000)
```

`igToPS` *Exports Interaction graph to a PS (PostScript) file*

Description

Exports Interaction graph to a PS (PostScript) file

Usage

```
igToPS(ig, path = "", fName = "InteractionGraph", h = 2000)
```

Arguments

`ig`	Interaction graph
`path`	The folder in which to write the PS file;
`fName`	The name of the file to be created; "InteractionGraph" by default
`h`	Desired height of the image in pixels; 2000px by default

Value

Writes the `ig` interaction graph to a PostScript (`.ps`) file to the folder specified in the `path`

Examples

```
#create temp dir path with slashes
myDir <- gsub("\\\\", "/", tempdir())

#create interaction graph
g <- interactionGraph(golf, "Play", intNo = 10)

#write to PS
igToPS(g, path = myDir, fName = "MyGraph", h = 2000)
```

igToSVG *Exports Interaction graph to a SVG file*

Description

Exports Interaction graph to a SVG file

Usage

```
igToSVG(ig, path = "", fName = "InteractionGraph", h = 2000)
```

Arguments

ig	Interaction graph
path	The folder in which to write the SVG file;
fName	The name of the file to be created; "InteractionGraph" by default
h	Desired height of the image in pixels; 2000px by default

Value

Writes the ig interaction graph to a SVG (.svg) file to the folder specified in the path

Examples

```
#create temp dir path with slashes
myDir <- gsub("\\\\", "/", tempdir())

#create interaction graph
g <- interactionGraph(golf, "Play", intNo = 10)

#write to SVG
igToSVG(g, path = myDir, fName = "MyGraph", h = 2000)
```

infoGain *Calculates Information Gain (2-way Interaction Gain) of a discrete data.frame*

Description

$InfoGAIN = H(S) - H(S|X)$, where $H(S)$ is the difference in the Shannon's entropy of the system S before a new attribute X is introduced, and $H(S|X)$ is the entropy of the system after the attribute X has been introduced.

Usage

```
infoGain(df, inAtt, classAtt)
```

Arguments

df	A discrete data.frame
inAtt	An input column of the data.frame df (string)
classAtt	A class column of the data.frame df (string)

Value

The Information Gain of df on the class attribute classAtt

Examples

```
infoGain(golf, "Windy", "Play")

infoGain(golf, "Outlook", "Play")
```

---

interactionGraph *Creates Interaction graph*

Description

Creates Interaction graph

Usage

```
interactionGraph(df, classAtt, intNo = 16, speedUp = FALSE)
```

Arguments

df	A discrete data.frame
classAtt	A class column of the df (string)
intNo	A desired number of interactions to show, i.e. an (integer) in range: [2,20]; Default value is 16.
speedUp	A (boolean) parameter. If TRUE, indicates whether the pairs of attributes with Information Gain equal to zero (on the 4th decimal) should be pruned. This speeds up calculations for larger datasets. By default it is turned off (i.e. set to FALSE).

Value

An interaction graph object (string)

Examples

```
interactionGraph(golf, "Play", intNo = 10)
interactionGraph(golf, "Play", intNo = 10, speedUp = FALSE)
interactionGraph(golf, "Play", intNo = 10, speedUp = TRUE)
```

interactions3Way *Calculates 3-Way Interactions*

Description

Formula: $I(X;Y;C) = I(X,Y;C) - IG(X;C) - IG(Y;C)$, where $I(X;Y;C)$ is 3-way Interaction gain of the attributes X and Y, given the context (i.e. class) attribute C. Hence, $I(X,Y;C)$ is a joint 2-way interaction gain (i.e. Information Gain) of the attributes X and Y, and $I(X;C)$ and $I(Y;C)$ are 2-way Interaction gains (i.e. Information Gains) of the attributes X and Y, respectively.

Usage

```
interactions3Way(df, classAtt, speedUp = FALSE)
```

Arguments

df	A discrete data.frame
classAtt	A class column of the df (string)
speedUp	A (boolean) parameter. If TRUE, indicates whether the pairs of attributes with Information Gain equal to zero (on the 4th decimal) should be pruned. This speeds up calculations for larger datasets. By default it is turned off (i.e. set to FALSE).

Value

A list with a: 1) data frame with 3-way interactions, 2)list of 2-way interactions of the input attributes

Examples

```
interactions3Way(golf, "Play")
interactions3Way(golf, "Play", speedUp = TRUE)
interactions3Way(golf, "Play", speedUp = FALSE)
```

isDiscreteDataFrame *Tests if data.frame is discrete (i.e. all of its columns are factors)*

Description

Tests if data.frame is discrete (i.e. all of its columns are factors)

Usage

```
isDiscreteDataFrame(df)
```

Arguments

df	A data.frame

Value

Boolean: TRUE if all columns of the data.frame df are factors, FALSE otherwise; If the provided df object is of other type than data.frame, the function throws an error.

Examples

```
isDiscreteDataFrame(golf)
```

---

plotIntGraph *Plots Interaction graph*

Description

Plots Interaction graph

Usage

```
plotIntGraph(ig)
```

Arguments

ig	Interaction graph

Value

Plots the ig

Examples

```
plotIntGraph(interactionGraph(golf, "Play", intNo = 10))
```

print.ig *Print generic method for Interaction Graph (S3 class)*

Description

Print generic method for Interaction Graph (S3 class)

Usage

```
## S3 method for class 'ig'
print(intGraph)
```

Arguments

intGraph	An (ig) object

Value

Print (ig) object

print.igEdge *Print generic method for Interaction Graph Edges (S3 class)*

Description

Print generic method for Interaction Graph Edges (S3 class)

Usage

```
## S3 method for class 'igEdge'
print(edge)
```

Arguments

edge	An (igEdge) object

Value

Print (igEdge) object

print.igNode *Print generic method for Interaction Graph Nodes (S3 class)*

Description

Print generic method for Interaction Graph Nodes (S3 class)

Usage

```
## S3 method for class 'igNode'
print(node)
```

Arguments

node	An (igNode) object

Value

Print (igNode) object

toString.ig *toString() generic method for Interaction Graph (S3 class)*

Description

toString() generic method for Interaction Graph (S3 class)

Usage

```
## S3 method for class 'ig'
toString(intGraph)
```

Arguments

intGraph	An ig object

Value

A character object made of the provided ig object

toString.igEdge *toString() generic method for Interaction Graph Edges (S3 class)*

Description

toString() generic method for Interaction Graph Edges (S3 class)

Usage

```
## S3 method for class 'igEdge'
toString(edge)
```

Arguments

edge	An (igEdge) object

Value

(character) object made of the provided (igEdge) object

toString.igNode *toString() generic method for Interaction Graph Nodes (S3 class)*

Description

toString() generic method for Interaction Graph Nodes (S3 class)

Usage

```
## S3 method for class 'igNode'
toString(node)
```

Arguments

node	An (igNode) object

Value

(character) object made of the provided (igNode) object

Index