Title Helper Functions for QCA in R

Version 0.2.3

Author Jirka Lewandowski <jirka.lewandowski@wzb.eu> [aut, cre]

Maintainer Jirka Lewandowski <jirka.lewandowski@wzb.eu>

Description Helper functions for Qualitative Comparative Analysis: evaluate and plot Boolean formulae on fuzzy set score data, apply Boolean operations, compute consistency and coverage measures.

Depends R (>= 3.1.0)

Imports stringr (>= 0.6.2), ggplot2 (>= 0.9.3.1), directlabels (>= 2013.6.15), graphics, QCA (>= 2.5)

License GPL (>= 3)

LazyData true

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-01-03 23:54:33

R topics documented:

and ................................................. 2
consistency ........................................ 3
evaluate_dnf ....................................... 4
format_dnf ......................................... 5
formula_to_function ............................... 5
plot.qca ......................................... 6
QCAtools ........................................... 6
xyplot ............................................ 7

Index 8
Description

Logical 'and' of two conditions
Logical 'or' of two conditions
Logical 'not' of a condition

Usage

\[
\begin{align*}
\text{and} & : v_1, v_2 \\
\text{or} & : v_1, v_2 \\
\text{not} & : v
\end{align*}
\]

Arguments

- \(v_1\): A vector of fuzzy set scores of cases
- \(v_2\): A vector of fuzzy set scores of cases
- \(v\): A vector of fuzzy set scores of cases

Value

- the fuzzy set scores of the logical conjunction of \(v_1\) and \(v_2\) for each case, i.e. the minimum in each component
- the fuzzy set scores of the logical disjunction of \(v_1\) and \(v_2\) for each case, i.e. the maximum in each component
- the fuzzy set scores of the negation of \(v\) for each case, i.e. 1-\(v\)

Examples

\[
\begin{align*}
\text{and}(c(0, 0.5, 1), c(0.25, 0.75, 0.75)) \\
\text{or}(c(0, 0.5, 1), c(0.25, 0.75, 0.75)) \\
\text{not}(c(0, 0.5, 1))
\end{align*}
\]
Compute the consistency value

**Description**

Computes the consistency score of "formula1 -> formula2" (sufficient condition) or "formula1 <- formula2" (necessary condition), depending on whether type is "->" or "<-". If type is "<->" it computes an equivalence score of formula1 and formula2 via the formula $\frac{\sum(\min(X,Y))}{\sum(\max(X,Y))}$

**Usage**

`consistency(formula1, type = "->", formula2, data)`

**Arguments**

- `formula1`: A string, list of strings or function representing a Boolean formula in disjunctive normal form
- `type`: either "->", "<-" or "<->", depending on the direction of the implication that is to be evaluated
- `formula2`: A string, list of strings or function representing a Boolean formula in disjunctive normal form
- `data`: A data frame where the rows represent cases and the columns the sets. Column names must be as in the formula.

**Details**

Compute a consistency score for an implication/necessity/sufficiency statement.

If `formula` is a function, it must take a data.frame and return a vector.

If `formula` is a string or list of strings, the following conventions hold:
- Set names must be capitalized in the formula and the data; if they are lowercase, they are interpreted as the negation of the set.
- If `formula` is a string, logical 'or' is expressed as a '+', and logical 'and' as a '*'.
- If `formula` is a list of strings, the strings are assumed to be the disjuncts and are concatenated with '+'. The formula must be in disjunctive normal form, i.e. it must be a disjunction of conjunctions of elementary or negated elementary sets. Example: `A*B*C + a*B`

**Value**

the consistency score of the implication described by `formula1`, `type` and `formula2`

**Examples**

```r
require(QCA)
data(d.urban)
consistency("MLC + FRB", "->", "CP", d.urban)
```
**evaluate_dnf**

**Evaluate a formula**

**Description**

When given a Boolean formula (see details) and a data.frame of cases and fuzzy set score for conditions, computes for each case the score of the membership in the set described by the formula.

**Usage**

```r
evaluate_dnf(data, formula)
```

**Arguments**

- `data`: A data frame where the rows represent cases and the columns the sets. Column names must be as in the formula.
- `formula`: A string, list of strings or function representing a Boolean formula in disjunctive normal form.

**Details**

If `formula` is a function, it must take a `data.frame` and return a vector.

If `formula` is a string or list of strings, the following conventions hold: Set names must be capitalized in the formula and the data; if they are lowercase, they are interpreted as the negation of the set. If `formula` is a string, logical 'or' is expressed as a '+' and logical 'and' as a '*'. If `formula` is a list of strings, the strings are assumed to be the disjuncts and are concatenated with '+'. The formula must be in disjunctive normal form, i.e. it must be a disjunction of conjunctions of elementary or negated elementary sets. Example: `A*B*C + A*B`

**Value**

the fuzzy set score of the set described by the formula for each case in the data.

**Examples**

```r
require(QCA)
data(d.urban)
evaluate_dnf(d.urban, "MLC*frb + CP")
```
format_dnf

Rewrite a list of clauses to a string containing a Boolean formula in disjunctive normal form

Description
Rewrite a list of clauses to a string containing a Boolean formula in disjunctive normal form

Usage
format_dnf(dnf)

Arguments

Arguments

dnf list of clauses

Value
string containing a Boolean formula in disjunctive normal form

formula_to_function
Convert formula to function

Description
When given a Boolean formula (in disjunctive normal form, see details), this function produces a function that takes a data.frame of a QCA data table and computes the fuzzy set score for each case of membership in the set described by the formula

Usage
formula_to_function(formula)

Arguments

Arguments

formula A string or vector of strings containing a Boolean formula in disjunctive normal form

Details
Set names must be capitalized in the formula and the data; if they are lowercase, they are interpreted as the negation of the set. If formula is a string, logical 'or' is expressed as a '+' and logical 'and' as a '*'. If formula is a list of strings, the strings are assumed to be the disjuncts and are concatenated with '+'. Disjunctive normal form means that the formula must be a disjunction of conjunctions of elementary or negated elementary sets. Example: a*b*c + a*b
a function that takes a data.frame and computes the fuzzy set score of the set described by the
formula for each case into a vector

Examples

formula_to_function("A*b*C + a*B")

plot.qca

Plot the fuzzy set scores of the solution and the outcome against each other

Description

Plot the fuzzy set scores of the solution and the outcome against each other

Usage

## S3 method for class 'qca'
plot(x, ...)

Arguments

x

an object of class qca as returned by eqmcc of the package QCA

... further arguments passed on to xyplot

Value

the ggplot object

Examples

## Not run:
require(QCA)
data(d.urban)
solution <- eqmcc(d.urban, outcome="RT", conditions=c("MLC", "FRB", "CP", "WSR"))
plot(solution)

## End(Not run)
**xyplot**

Plot fuzzy set score of two sets against each other

**Description**
Plot fuzzy set score of two sets against each other

**Usage**

```r
xyplot(x, y, data, labels = FALSE, main.diagonal = TRUE,
        anti.diagonal = FALSE)
```

**Arguments**

- **x**: Formula that describes the fuzzy set to plot along the x axis
- **y**: Formula that describes the fuzzy set to plot along the y axis
- **data**: Data set of basic fuzzy set scores
- **labels**: flag whether to label individual points with the case names
- **main.diagonal**: flag whether to plot the main diagonal
- **anti.diagonal**: flag whether to plot the anti diagonal

**Value**
the ggplot plot object

**Examples**

```r
require(QCA)
data(d.urban)
xyplot("MLC", "WSR", d.urban)
```
Index

and, 2
consistency, 3
eqmcc, 6
evaluate_dnf, 4
format_dnf, 5
formula_to_function, 5
not (and), 2
or (and), 2
plot.qca, 6
QCAtools, 6
QCAtools-package (QCAtools), 6
xyplot, 6, 7