

Package: mcbiopi (via r-universe)

September 13, 2024

Title Matrix Computation Based Identification of Prime Implicants

Version 1.1.6

Date 2018-10-06

Author Holger Schwender

Maintainer Holger Schwender <holger.schw@gmx.de>

Imports methods

Description Computes the prime implicants or a minimal disjunctive normal form for a logic expression presented by a truth table or a logic tree. Has been particularly developed for logic expressions resulting from a logic regression analysis, i.e. logic expressions typically consisting of up to 16 literals, where the prime implicants are typically composed of a maximum of 4 or 5 literals.

License LGPL (>= 2)

NeedsCompilation no

Date/Publication 2018-10-06 20:10:03 UTC

Repository <https://holgerschw.r-universe.dev>

RemoteUrl <https://github.com/cran/mcbiopi>

RemoteRef HEAD

RemoteSha 686ec565ed192fb7dc5d3c98399334792fe95aa5

Contents

generateTruthTab	2
minDNF	2
prime.implicants	3

Index	5
--------------	----------

generateTruthTab *Truth Table for a Logic Tree*

Description

Generates the truth table or the prime implicants, respectively, for a logic tree built in a logic regression,

Usage

```
generateTruthTab(ltree)

getPImps(ltree, type)
```

Arguments

ltree an object of class logregtree.
type the type of the logic regression model that has been fitted.

Author(s)

Holger Schwender, <holger.schwender@hhu.de>

See Also

[prime.implicants](#)

minDNF *Minimum Disjunctive Normal Form*

Description

Computes the minimal disjunctive normal form for a given truth table.

Usage

```
minDNF(mat)
```

Arguments

mat a matrix containing only 0's and 1's. Each column of mat corresponds to a binary variable and each row to a combination of the variables for which the logic expression is TRUE.

Value

An object of class `minDNF` containing a vector comprising a minimized set of prime implicants. If more than one solution exist, then a list is returned containing all solutions.

Author(s)

Holger Schwender, <holger.schwender@hhu.de>

References

Schwender, H. (2007). Minimization of Boolean Expressions Using Matrix Algebra. Technical Report, SFB 475, Department of Statistics, TU Dortmund University.

See Also

[prime.implicants](#)

Examples

```
## Not run:
# Generate the truth table considered in Schwender (2007).

mat <- matrix(c(rep(0, 4), rep(1, 6),
  rep(0, 6), rep(1, 4),
  0, 0, 1, 1, 0, 1, 0, 0, 1, 1,
  0, 1, 0, 1, 1, 1, 0, 1, 0, 1), ncol=4)
colnames(mat) <- paste("X", 1:4, sep="")

# Computing the minimal disjunctive normal form.

minDNF(mat)

## End(Not run)
```

prime.implicants	<i>Prime Implicants</i>
------------------	-------------------------

Description

Computes the prime implicants of a given truth table.

Usage

```
prime.implicants(mat)
```

Arguments

`mat` a matrix containing only 0's and 1's. Each column of `mat` corresponds to a binary variable and each row to a combination of the variables for which the logic expression is TRUE.

Value

An object of class `primeImp` containing a vector `vec.primes` comprising the prime implicants and a matrix `mat.primes` representing the prime implicant table.

Author(s)

Holger Schwender, <holger.schwender@hhu.de>

References

Schwender, H. (2007). Minimization of Boolean Expressions Using Matrix Algebra. Technical Report, SFB 475, Department of Statistics, TU Dortmund University.

See Also

[minDNF](#)

Examples

```
## Not run:
# Generate the truth table considered in Schwender (2007).

mat <- matrix(c(rep(0, 4), rep(1, 6),
  rep(0, 6), rep(1, 4),
  0, 0, 1, 1, 0, 1, 0, 0, 1, 1,
  0, 1, 0, 1, 1, 1, 0, 1, 0, 1), ncol=4)
colnames(mat) <- paste("X", 1:4, sep="")

# Determining the prime implicants.

prime.implicants(mat)

## End(Not run)
```

Index

- * **logic**
 - generateTruthTab, 2
 - minDNF, 2
 - prime.implicants, 3
- * **optimize**
 - minDNF, 2
- * **print**
 - minDNF, 2
 - prime.implicants, 3
- * **utilities**
 - generateTruthTab, 2

evalTree (generateTruthTab), 2

generateTruthTab, 2

getPImps (generateTruthTab), 2

minDNF, 2, 4

prime.implicants, 2, 3, 3

print.minDNF (minDNF), 2

print.primeImp (prime.implicants), 3