Package: FuzzyM (via r-universe)

May 26, 2026

Type Package

Title Fuzzy Cognitive Maps Operations

Version 0.1.0

Author Alina Petukhova

Maintainer Alina Petukhova <petukhova.alina@gmail.com>

Description Contains functions for operations with fuzzy cognitive maps using t-norm and s-norm operators. T-norms and S-norms are described by Dov M. Gabbay and George Metcalfe (2007) <doi:10.1007/s00153-007-0047-1>. System indicators are described by Cox, Earl D. (1995) <isbn:1886801010>. Executable examples are provided in the ``inst/examples" folder.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.2

Imports utils

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Repository https://petukhovaalina-tech.r-universe.dev

Date/Publication 2022-02-22 07:20:08 UTC

RemoteUrl https://github.com/cran/FuzzyM

RemoteRef HEAD

RemoteSha 40ce255e30aafd970338f30e4906f087d3d91537

Contents

---

maxtix_tranz *matrix_tranz*

Description

The maxtix_tranz set of functions is aimed to calculate dissonance, consonance and influence

Usage

```
tnorm_functions

snorm_functions

snorm_functions_reverse

tnorm_functions_reverse

positive_matrix_calc(initmatrix)

transitive_closure(
  positivematrix,
  tnorm,
  snorm,
  snormMatrix,
  gammaTnormMean,
  algaTnorm,
  gammaTnorm,
  piTnorm,
  gammaSnorm,
  piSnorm
)

matrix_transitive_join(matrix, snorm, gammaSnorm, piSnorm)

consonanse_dissonanse(finalmatrix)

cross_consonanse(finalmatrix)

cross_dissonanse(finalmatrix)

cross_positive_influence(finalmatrix)
```

```
cross_negative_influence(finalmatrix)

impuls_vector(vector, matrix)

multiply_vector(matrix, vector)

multiply_matrix(
  matrix_1,
  matrix_2,
  tnorm,
  snorm,
  gammaTnormMean,
  algaTnorm,
  gammaTnorm,
  piTnorm,
  gammaSnorm,
  piSnorm
)

maximum_matrix(matrix_1, matrix_2)

ik_pos_maximum(matrix, initMatrix, ipath, jpath)

ik_neg_maximum(matrix, initMatrix, ipath, jpath)

reverse_task(
  df_matrix,
  vectorY,
  tnorm,
  tnorm_reverse,
  snorm,
  snormMatrix,
  snorm_reverse
)

direct_task(df_matrix, vectorX, tnorm, snorm, snormMatrix)
```

Arguments

initmatrix	matrix
positivematrix	matrix
tnorm	function
snorm	function
snormMatrix	function
gammaTnormMean	function
algaTnorm	function
gammaTnorm	function

piTnorm	function
gammaSnorm	function
piSnorm	function
matrix	matrix
finalmatrix	matrix
vector	matrix
matrix_1	matrix
matrix_2	matrix
initMatrix	matrix
ipath	vector
jpath	vector
df_matrix	matrix
vectorY	vector
tnorm_reverse	function
snorm_reverse	function
vectorX	vector

Value

eigen values of initmatrix

positive matrix of initmatrix

transitive closure of positivematrix

aggregation function for transitive closure of matrix

system indicators of finalmatrix

cross consonanse of finalmatrix

cross dissonanse of finalmatrix

cross positive influence of finalmatrix

cross negative influence of finalmatrix

impulse of matrix based on vector

multiplication of matrix and vector

multiplication of matrix_1 and matrix_2

maximum of matrix_1 and matrix_2

ik walk for matrix based on initMatrix with ipath and jpath

ik negative walk for matrix based on initMatrix with ipath and jpath

reverse task solution for df_matrix with vectorY using tnorm, tnorm_reverse, snorm, snormMatrix, snorm_reverse

direct task solution for df_matrix with vectorX using tnorm, snorm, snormMatrix

maxtix_tranz_probability

FCM package with functions for matrix probability calculations

Description

The maxtix_tranz_probability set of functions is aimed to calculate maximum matrix based on transitive closure

Usage

```
multiply_matrix_prob(matrix_1, matrix_2)

transitive_closure_prob(positivematrix)

transitive_closure_prob_max(positivematrix)

probability_matrix_transitive(maxmatrix)
```

Arguments

matrix_1	matrix
matrix_2	matrix
positivematrix	matrix
maxmatrix	matrix

Value

multiplication of matrix `matrix_1` and `matrix_2`

transitive closure of matrix `positivematrix`

transitive closure of matrix `positivematrix` with max function

joined transitive closure of matrix `maxmatrix`

polinomial_eq_dominant

FCM package with functions for matrix manipulations

Description

polinomial_eq_dominant set contains 1 function: reverse task calculation. Each function takes a matrix, vector and t-norm as arguments and returns a solution matrix. The calculation procedure of the function includes a solution existence check and a solution check.

Usage

```
calc_reverse_task(matrix, vector, tnorm, tnorm_reverse, snorm, snorm_reverse)
```

Arguments

matrix	matrix
vector	vector
tnorm	function
tnorm_reverse	function
snorm	function
snorm_reverse	function

Value

solution of polynomial equation of matrix matrix, vector using tnorm, tnorm_reverse, snorm, snorm_reverse

---

s_norm *s_norm*

Description

s_norm set of functions is aimed to calculate drastic, einstein, algebraic, hamacher products, min and bounded difference S-norms

Usage

```
drastic_sum_snorm(element1, element2, gammaSnorm, piSnorm)

bounded_sum_snorm(element1, element2, gammaSnorm, piSnorm)

einstein_sum_snorm(element1, element2, gammaSnorm, piSnorm)

algebraic_sum_snorm(element1, element2, gammaSnorm, piSnorm)

hamacher_sum_snorm(element1, element2, gammaSnorm, piSnorm)

max_snorm(element1, element2, gammaSnorm, piSnorm)

hamacher_union_operator_snorm(element1, element2, gammaSnorm, piSnorm)

yager_union_operator_snorm(element1, element2, gammaSnorm, piSnorm)

snorm_functions

get_snorm(typeSnorm)
```

Arguments

element1, element2
: paramater

gammaSnorm, piSnorm, typeSnorm
: norm

Format

An object of class list of length 8.

---

s_norm_reverse *FCM package with functions for reverse S-norms calculations*

Description

s_norm_reverse set of functions is aimed to calculate drastic, einstein, algebraic, hamacher products, min and bounded difference reverse S-norms

Usage

```
drastic_sum_snorm_reverse(element1, element2, gammaSnorm, piSnorm)

bounded_sum_snorm_reverse(element1, element2, gammaSnorm, piSnorm)

einstein_sum_snorm_reverse(element1, element2, gammaSnorm, piSnorm)

algebraic_sum_snorm_reverse(element1, element2, gammaSnorm, piSnorm)

hamacher_sum_snorm_reverse(element1, element2, gammaSnorm, piSnorm)

max_snorm_reverse(element1, element2, gammaSnorm, piSnorm)

hamacher_union_operator_snorm_reverse(element1, element2, gammaSnorm, piSnorm)

yager_union_operator_snorm_reverse(element1, element2, gammaSnorm, piSnorm)

snorm_functions_reverse

get_snorm_reverse(typeSnorm)
```

Arguments

element1, element2
: paramater

gammaSnorm, piSnorm, typeSnorm
: norm

Format

An object of class `list` of length 8.

`t_norm` *t_norm*

Description

t_norm set of functions is aimed to calculate drastic, einstein, algebraic, hamacher products, min and bounded difference T-norms

Usage

```
min_tnorm(element1, element2, gammaTnormMean, algaTnorm, gammaTnorm, piTnorm)

hamacher_product_tnorm(
  element1,
  element2,
  gammaTnormMean,
  algaTnorm,
  gammaTnorm,
  piTnorm
)

algebraic_product_tnorm(
  element1,
  element2,
  gammaTnormMean,
  algaTnorm,
  gammaTnorm,
  piTnorm
)

einstein_product_tnorm(
  element1,
  element2,
  gammaTnormMean,
  algaTnorm,
  gammaTnorm,
  piTnorm
)

bounded_difference_tnorm(
  element1,
  element2,
  gammaTnormMean,
  algaTnorm,
```

```
  gammaTnorm,
  piTnorm
)

drastic_product_tnorm(
  element1,
  element2,
  gammaTnormMean,
  algaTnorm,
  gammaTnorm,
  piTnorm
)

parameterized_mean_intersection_operator_tnorm(
  element1,
  element2,
  gammaTnormMean,
  algaTnorm,
  gammaTnorm,
  piTnorm
)

dubois_intersection_operator_tnorm(
  element1,
  element2,
  gammaTnormMean,
  algaTnorm,
  gammaTnorm,
  piTnorm
)

hamacher_intersection_operator_tnorm(
  element1,
  element2,
  gammaTnormMean,
  algaTnorm,
  gammaTnorm,
  piTnorm
)

yager_intersection_operator_tnorm(
  element1,
  element2,
  gammaTnormMean,
  algaTnorm,
  gammaTnorm,
  piTnorm
)
```

```
tnorm_functions

get_tnorm(typeTnorm)
```

Arguments

element1, element2
: paramater

gammaTnormMean, algaTnorm, gammaTnorm, piTnorm, typeTnorm
: norm

Format

An object of class list of length 10.

---

t_norm_reverse *t_norm_reverse*

Description

t_norm_reverse set of functions is aimed to calculate drastic, einstein, algebraic, hamacher products, min and bounded difference reverse T-norms

Usage

```
min_tnorm_reverse(
  element1,
  element2,
  gammaTnormMean,
  algaTnorm,
  gammaTnorm,
  piTnorm
)

hamacher_product_tnorm_reverse(
  element1,
  element2,
  gammaTnormMean,
  algaTnorm,
  gammaTnorm,
  piTnorm
)

algebraic_product_tnorm_reverse(
  element1,
  element2,
```

```
  gammaTnormMean,
  algaTnorm,
  gammaTnorm,
  piTnorm
)

einstein_product_tnorm_reverse(
  element1,
  element2,
  gammaTnormMean,
  algaTnorm,
  gammaTnorm,
  piTnorm
)

bounded_difference_tnorm_reverse(
  element1,
  element2,
  gammaTnormMean,
  algaTnorm,
  gammaTnorm,
  piTnorm
)

drastic_product_tnorm_reverse(
  element1,
  element2,
  gammaTnormMean,
  algaTnorm,
  gammaTnorm,
  piTnorm
)

parameterized_mean_intersection_operator_tnorm_reverse(
  element1,
  element2,
  gammaTnormMean,
  algaTnorm,
  gammaTnorm,
  piTnorm
)

dubois_intersection_operator_tnorm_reverse(
  element1,
  element2,
  gammaTnormMean,
  algaTnorm,
  gammaTnorm,
```

```
  piTnorm
)

hamacher_intersection_operator_tnorm_reverse(
  element1,
  element2,
  gammaTnormMean,
  algaTnorm,
  gammaTnorm,
  piTnorm
)

yager_intersection_operator_tnorm_reverse(
  element1,
  element2,
  gammaTnormMean,
  algaTnorm,
  gammaTnorm,
  piTnorm
)

tnorm_functions_reverse

get_tnorm_reverse(typeTnorm)
```

Arguments

`element1`, `element2`
: paramater

`gammaTnormMean`, `algaTnorm`, `gammaTnorm`, `piTnorm`, `typeTnorm`
: norm

Format

An object of class `list` of length 10.

Index