Package 'ZEP'

June 23, 2025

Type Package

Title Procedures Related to the Zadeh's Extension Principle for Fuzzy Data

Version 0.1.5

Description Procedures for calculation, plotting, and approximation of the outputs for fuzzy numbers (see A.I. Ban, L. Coroianu, P. Grzegorzewski ``Fuzzy Numbers: Approximations, Ranking and Applications" (2015)) based on the Zadeh's Extension Principle (see de Barros, L.C., Bassanezi, R.C., Lodwick, W.A. (2017) <doi:10.1007/978-3-662-53324-6_2>).

License GPL-3

NeedsCompilation no

Imports FuzzyNumbers, methods

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Author Maciej Romaniuk [cre, aut] (ORCID: <https://orcid.org/0000-0001-9649-396X>, (Systems Research Institute Polish Academy of Sciences, Newelska 6, 01-447 Warszawa, Poland)),
Abbas Parchami [aut] (ORCID: <https://orcid.org/0000-0002-0593-7324>, (Mahani Math Center, Afzalipour Research Institute, Shahid Bahonar University of Kerman, Kerman, Iran)),
Przemyslaw Grzegorzewski [ctb] (ORCID: <https://orcid.org/0000-0002-5191-4123>, (Faculty of Mathematics and Information Science, Warsaw University of Technology, Koszykowa 75, 00-662 Warsaw, Poland))

Maintainer Maciej Romaniuk <mroman@ibspan.waw.pl>

Repository CRAN

Date/Publication 2025-06-23 03:50:02 UTC

Contents

`ApplyZFunction` *Function to apply the Zadeh's principle*

Description

`ApplyZFunction` applies the selected function to a fuzzy number using the Zadeh's principle.

Usage

```
ApplyZFunction(
  value,
  FUN,
  knots = 10,
  approximation = FALSE,
  method = "NearestEuclidean",
  ...
)
```

Arguments

`value`	Input fuzzy number.
`FUN`	Function used for the input fuzzy number with the help of the Zadeh's principle.
`knots`	Number of the alpha-cuts used during calculation of the output.
`approximation`	If `TRUE`, the approximated output is calculated.
`method`	The selected approximation method.
`...`	Additional parameters passed to other functions.

Details

The function takes the input fuzzy number `value` (which should be described by one of the classes from `FuzzyNumbers` package) and applies the function `FUN` using the Zadeh's principle. The output is given by a fuzzy number or its approximation (when `approximation` is set to `TRUE` and the respective `method` is selected). To properly find the output, value of `FUN` is calculated for many alpha-cuts of `value`. The number of these alpha-cuts is equal to `knots` (plus 2 for the support and the core).

The input fuzzy number `value` should be given by fuzzy number described by classes from `FuzzyNumbers` package.

Value

The output is a fuzzy number described by classes from `FuzzyNumbers` package (piecewise linear fuzzy number without approximation, various types with the approximation applied).

Examples

```
library(FuzzyNumbers)

# prepare complex fuzzy number

A <- FuzzyNumber(-5, 3, 6, 20, left=function(x)
pbeta(x,0.4,3),
right=function(x) 1-x^(1/4),
lower=function(alpha) qbeta(alpha,0.4,3),
upper=function(alpha) (1-alpha)^4)

# find the output via the Zadeh's principle

ApplyZFunction(A,FUN=function(x)x^3+2*x^2-1)

# find the approximated output via the Zadeh's principle

ApplyZFunction(A,FUN=function(x)x^3+2*x^2-1,approximation=TRUE)
```

`approximationMehodsInside` *A vector containing names of the built-in approximation methods.*

Description

'approximationMehodsInside' is a vector containing names of the built-in approximation methods.

Usage

```
approximationMehodsInside
```

Format

An object of class `character` of length 2.

Value

This function returns a vector of strings.

Examples

```
# check the names

approximationMehodsInside
```

DpqDistance *Function to calculate D(p,q) distance.*

Description

DpqDistance calculates the generalized D(p,q) distance between two fuzzy numbers.

Usage

```
DpqDistance(value1, value2, p = 2, q = 1/2)
```

Arguments

value1	First fuzzy number.
value2	Second fuzzy number.
p	Value of the power (and the the root) applied in the distance calculation.
q	Value of the weight for the second fuzzy number (for the first one this weight is calculated as 1-q, respectively).

Details

The function calculates the generalized D(p,q) distance between two fuzzy numbers value1 and value2, where p is the value of the applied power, and q is the weight between these two fuzzy numbers.

All of the input values should be given by fuzzy numbers described by classes from FuzzyNumbers package.

Value

The output is a numerical value (the calculated distance).

Examples

```
library(FuzzyNumbers)

# prepare two fuzzy numbers

A <- TrapezoidalFuzzyNumber(0,1,2,3)

B <- TrapezoidalFuzzyNumber(1,3,4,6)
```

```
# calculate the distance

DpqDistance (A,B)
```

`FuzzyApproximation` *Function for approximation with the help of methods other than in FuzzyNumbers package*

Description

`FuzzyApproximation` approximates the given fuzzy number.

Usage

```
FuzzyApproximation(value, method = "ExpectedValueCore", ...)
```

Arguments

`value`	Fuzzy number to approximate.
`method`	The selected approximation method.
`...`	Additional parameters passed to other functions (like approximation method from the FuzzyNumbers package).

Details

The function approximates the fuzzy number given by `value` with the method selected by `method`. The following approximations are possible: `ExpectedValueCore`–preserving the expected value and the core of `value`, `TriangularSupportDist`–constructs the triangular fuzzy number based on minimization of `DpqDistance`, preserving the support of `value`, and the approximation methods from the FuzzyNumbers package (namely: `Naive`, `NearestEuclidean`, `ExpectedIntervalPreserving`, `SupportCoreRes`

The input value should be given by a fuzzy number described by classes from `FuzzyNumbers` package.

Value

The output is a fuzzy number (triangular or trapezoidal one) described by classes from `FuzzyNumbers` package.

Examples

```
library(FuzzyNumbers)

# prepare complex fuzzy number

A <- FuzzyNumber(-5, 3, 6, 20, left=function(x)
pbeta(x,0.4,3),
right=function(x) 1-x^(1/4),
lower=function(alpha) qbeta(alpha,0.4,3),
upper=function(alpha) (1-alpha)^4)

# find approximation

FuzzyApproximation (A)
```

`PlotZFunction` *Plot input and output for the Zadeh's principle*

Description

`PlotZFunction` applies the selected function to a fuzzy number using the Zadeh's principle, and plots the input and output.

Usage

```
PlotZFunction(
  value,
  FUN,
  knots = 10,
  grid = TRUE,
  alternate = FALSE,
  approximation = FALSE,
  method = "NearestEuclidean",
  ...
)
```

Arguments

`value`	Input fuzzy number.
`FUN`	Function used for the input fuzzy number with the help of the Zadeh's principle.
`knots`	Number of the alpha-cuts used during calculation of the output.
`grid`	If `TRUE`, then additional grid is plotted.
`alternate`	If `TRUE`, the second type of the layout of figures is used.
`approximation`	If `TRUE`, the approximated output is calculated.
`method`	The selected approximation method.
`...`	Additional parameters passed to other functions.

Details

The function takes the input fuzzy number `value` (which should be described by one of the classes from `FuzzyNumbers` package) and applies the function `FUN` using the Zadeh's principle. The output is given by a fuzzy number or its approximation (when `approximation` is set to `TRUE` and the respective `method` is selected). To properly find the output, value of `FUN` is calculated for many alpha-cuts of `value`. The number of these alpha-cuts is equal to `knots` (plus 2 for the support and the core). The input and output fuzzy numbers are plotted together with the applied function. If the approximation is used, then also the approximated fuzzy number is shown (green line).

The input fuzzy number `value` should be given by fuzzy number described by classes from `FuzzyNumbers` package.

Value

Three (or four) figures are plotted: the input fuzzy number, the respective output (for the Zadeh's principle and the applied function), and the function. The output fuzzy number can be approximated with the selected method and also plotted.

Examples

```
library(FuzzyNumbers)

# prepare complex fuzzy number

A <- FuzzyNumber(-5, 3, 6, 20, left=function(x)
pbeta(x,0.4,3),
right=function(x) 1-x^(1/4),
lower=function(alpha) qbeta(alpha,0.4,3),
upper=function(alpha) (1-alpha)^4)

# plot the figures

PlotZFunction(A,FUN=function(x)x^3+2*x^2-1)

# find and plot the approximated output via the Zadeh's principle

PlotZFunction(A,FUN=function(x)x^3+2*x^2-1,approximation=TRUE)
```

Index