Package ‘FuzzyNumbers’

February 15, 2013

License LGPL (>= 3)

Maintainer Marek Gagolewski <gagolews@ibspan.waw.pl>

BugReports http://r-forge.r-project.org/tracker/?atid=5289&group_id=1408

LazyData true

Title Tools to deal with fuzzy numbers in R

Type Package

Author Marek Gagolewski <gagolews@ibspan.waw.pl>

Description The FuzzyNumbers package provides S4 classes and methods
to deal with Fuzzy Numbers. It allows for computations of
arithmetic operations on FNs, approximation by trapezoidal and
piecewise linear FNs, random FN generation, etc.

Version 0.02

URL http://www.ibspan.waw.pl/~gagolews/FuzzyNumbers/

Date 2012-12-27

Depends R (>= 2.15.0), graphics, methods

Suggests testthat


Repository CRAN

Date/Publication 2012-12-27 14:33:03

NeedsCompilation no
### \textbf{R topics documented:}

<table>
<thead>
<tr>
<th>Topics</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>FuzzyNumbers-package</td>
<td>3</td>
</tr>
<tr>
<td>alphacut</td>
<td>4</td>
</tr>
<tr>
<td>alphaInterval</td>
<td>5</td>
</tr>
<tr>
<td>ambiguity</td>
<td>6</td>
</tr>
<tr>
<td>approx.invert</td>
<td>6</td>
</tr>
<tr>
<td>Arithmetic</td>
<td>7</td>
</tr>
<tr>
<td>as.FuzzyNumber</td>
<td>8</td>
</tr>
<tr>
<td>as.PiecewiseLinearFuzzyNumber</td>
<td>8</td>
</tr>
<tr>
<td>convert.alpha</td>
<td>9</td>
</tr>
<tr>
<td>convert.side</td>
<td>9</td>
</tr>
<tr>
<td>core</td>
<td>10</td>
</tr>
<tr>
<td>DiscontinuousFuzzyNumber</td>
<td>10</td>
</tr>
<tr>
<td>DiscontinuousFuzzyNumber-class</td>
<td>12</td>
</tr>
<tr>
<td>distance</td>
<td>12</td>
</tr>
<tr>
<td>evaluate</td>
<td>13</td>
</tr>
<tr>
<td>expectedInterval</td>
<td>14</td>
</tr>
<tr>
<td>expectedValue</td>
<td>15</td>
</tr>
<tr>
<td>Extract</td>
<td>15</td>
</tr>
<tr>
<td>FuzzyNumber</td>
<td>16</td>
</tr>
<tr>
<td>FuzzyNumber-class</td>
<td>17</td>
</tr>
<tr>
<td>integrateAlpha</td>
<td>19</td>
</tr>
<tr>
<td>integrate_discont_val</td>
<td>20</td>
</tr>
<tr>
<td>piecewiseLinearApproximation</td>
<td>20</td>
</tr>
<tr>
<td>PiecewiseLinearFuzzyNumber</td>
<td>21</td>
</tr>
<tr>
<td>PiecewiseLinearFuzzyNumber-class</td>
<td>22</td>
</tr>
<tr>
<td>plot</td>
<td>23</td>
</tr>
<tr>
<td>PowerFuzzyNumber</td>
<td>24</td>
</tr>
<tr>
<td>PowerFuzzyNumber-class</td>
<td>25</td>
</tr>
<tr>
<td>show</td>
<td>25</td>
</tr>
<tr>
<td>supp</td>
<td>26</td>
</tr>
<tr>
<td>trapezoidalApproximation</td>
<td>26</td>
</tr>
<tr>
<td>TrapezoidalFuzzyNumber</td>
<td>28</td>
</tr>
<tr>
<td>TrapezoidalFuzzyNumber-class</td>
<td>28</td>
</tr>
<tr>
<td>value</td>
<td>29</td>
</tr>
<tr>
<td>weightedExpectedValue</td>
<td>29</td>
</tr>
<tr>
<td>width</td>
<td>30</td>
</tr>
</tbody>
</table>

**Index** 31
FuzzyNumbers-package

Tools to deal with fuzzy numbers in R

Description

**FuzzyNumbers** is an open source (LGPL 3) package for R. It provides S4 classes and methods to deal with Fuzzy Numbers and allows for computations of arithmetic operations on FNs, approximation by trapezoidal and piecewise linear FNs, random fuzzy numbers generation [TO DO] etc. The package may be used by the practitioners as well as by the researchers in fuzzy numbers theory (e.g. for testing new algorithms, generating numerical examples, preparing figures).

Details

Fuzzy set theory lets us effectively and quite intuitively represent imprecise or vague information. Fuzzy numbers, which form a particular subclass of fuzzy sets of the real line, play a significant role in many important theoretical and/or practical considerations. This is because we often describe our knowledge about objects through numbers, e.g. "I’m about 180 cm tall" or "The rocket was launched between 2 and 3 p.m."

For the formal definition of a fuzzy number please refer to the [FuzzyNumber man page](http://www.ibspan.waw.pl/~gagolews). Note that this package also deals with particular types of fuzzy numbers like trapezoidal, piecewise linear, or “parametric” FNs (see [TrapezoidalFuzzyNumber](http://www.ibspan.waw.pl/~gagolews/FuzzyNumbers/doc/FuzzyNumbers-Tutorial.pdf), [PiecewiseLinearFuzzyNumber](http://www.ibspan.waw.pl/~gagolews/FuzzyNumbers/doc/FuzzyNumbers-Tutorial.pdf), [PowerFuzzyNumber](http://www.ibspan.waw.pl/~gagolews/FuzzyNumbers/doc/FuzzyNumbers-Tutorial.pdf), [DiscontinuousFuzzyNumber](http://www.ibspan.waw.pl/~gagolews/FuzzyNumbers/doc/FuzzyNumbers-Tutorial.pdf))

The package aims to provide the following functionality:

1. Representation of arbitrary fuzzy numbers (including FNs with discontinuous side functions and/or alpha-cuts), as well as their particular types, e.g. trapezoidal and piecewise linear fuzzy numbers,
2. Defuzzification and Approximation by Triangular and Piecewise Linear FNs (see e.g. `expectedValue`, `value`, `trapezoidalApproximation`, `piecewiseLinearApproximation`),
3. Visualization of FNs (see `plot`),
4. Operations on FNs [TO DO],
5. Aggregation of FNs [TO DO],
6. Ranking of FNs [TO DO],
7. Random FN generation [TO DO],
8. …

Please feel free to send any comments and feature requests to the author (see his homepage at [http://www.ibspan.waw.pl/~gagolews](http://www.ibspan.waw.pl/~gagolews)).


**Keywords**: Fuzzy Numbers, Fuzzy Sets, Shadowed Sets, Trapezoidal Approximation, Piecewise Linear Approximation, Approximate Reasoning, Imprecision, Vagueness, Randomness.
Author(s)

Marek Gagolewski <gagolews@ibspan.waw.pl>

References


---

**alphacut**

*Calculate given alpha-cuts*

**Description**

Calculate given alpha-cuts

**Value**

a matrix with two columns or a vector of length two
Methods

signature(object = "FuzzyNumber", alpha = "numeric")

See Also

Other FuzzyNumber.method: alphaInterval, alphaInterval, alphaInterval, alphaInterval, ambiguity, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, convert.side, core, distance, distance, distance, distance, evaluate, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, Extract, Extract, Extract, Extract, Extract, FuzzyNumber-class, integrateAlpha, integrateAlpha, piecewiseLinearApproximation, plot, plot, plot, show, show, show, show, show, supp, trapezoidalApproximation, value, weightedExpectedValue, width

alphaInterval (numeric integration used)

signature(object = "TrapezoidalFuzzyNumber") (exact)

signature(object = "PiecewiseLinearFuzzyNumber") (exact)

signature(object = "PowerFuzzyNumber") (exact)

See Also

integrateAlpha

Other FuzzyNumber.method: alphacut, ambiguity, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, convert.side, core, distance, distance, distance, distance, distance, evaluate, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, Extract, Extract, Extract, Extract, Extract, FuzzyNumber-class, integrateAlpha, integrateAlpha, integrateAlpha, piecewiseLinearApproximation, plot, plot, plot, show, show, show, show, supp, trapezoidalApproximation, value, weightedExpectedValue, width

Description

We have $\alpha - Int(A) := [\int_0^1 \alpha A_L(\alpha) \, d\alpha, \int_0^1 \alpha A_U(\alpha) \, d\alpha]$.

Details

Note that this may be done with numeric integration (for instances of the FuzzyNumber and DiscontinuousFuzzyNumber class)

Methods

signature(object = "FuzzyNumber") (numerical integration used)

signature(object = "TrapezoidalFuzzyNumber") (exact)

signature(object = "PiecewiseLinearFuzzyNumber") (exact)

signature(object = "PowerFuzzyNumber") (exact)
ambiguity

Calculate the ambiguity of a fuzzy number

Description

The ambiguity is a measure of nonspecificity of a fuzzy number. It is defined as
\[ \text{amb}(A) := \int_0^1 \alpha (A_U(\alpha) - A_L(\alpha)) \, d\alpha. \]

Methods

signature(object = "FuzzyNumber")

References


See Also

alphaInterval on which the method is based

approx.invert

Approximate the inverse of a given side generating functions using interpolation

Description

The function is a wrapper to splinefun() and approxfun(). It may be used to create side generating functions from alpha-cut generators and inversely.

Usage

approx.invert(f, method = c("monoH.FC", "linear", "hyman", n = 500)
Arguments

\( f \) 
- a monotonic, continuous function \( f : [0,1] \rightarrow [0,1] \)

\( \text{method} \) 
- interpolation method: "monoH.FC", "hyman" or "linear"

\( n \) 
- number of interpolation points

Value

a new function, the approximate inverse of the input

See Also

FuzzyNumber
**as.FuzzyNumber**

*Converts a trapezoidal or a piecewise linear fuzzy number object to a (general) FuzzyNumber*

**Description**

FuzzyNumber is the base class for all FNs. Note that some functions for TFNs or PLFNs (more specific FNs) work much faster and are more precise. This function shouldn’t be used in normal computations.

**Usage**

```
as.FuzzyNumber(object)
```

**Arguments**

- **object**: a trapezoidal or piecewiselinear fuzzy number

**Value**

Object of class FuzzyNumber

**See Also**

`FuzzyNumber-class`, `TrapezoidalFuzzyNumber-class`, `PiecewiseLinearFuzzyNumber-class`

---

**as.PiecewiseLinearFuzzyNumber**

*Converts a trapezoidal fuzzy number object to a piecewise linear fuzzy number*

**Description**

Converts a trapezoidal fuzzy number object to a piecewise linear fuzzy number

**Usage**

```
as.PiecewiseLinearFuzzyNumber(object, knot.n = 0, knot.alpha = numeric(0))
```

**Arguments**

- **object**: a trapezoidal fuzzy number
- **knot.n**: the number of knots
- **knot.alpha**: knot.n alpha-cut values at knots
Value

Object of class PiecewiseLinearFuzzyNumber

Description

The resulting function calls the original function and then linearly scales its output.

Usage

convert.alpha(f, y1, y2)

Arguments

f a function into [y1,y2]
y1 numeric vector of length 1
y2 numeric vector of length 1

Value

a new function defined on [0,1] (scaled input)

See Also

FuzzyNumber

convert.side

Convert a given side function to side generating function

Description

The resulting function linearly scales the input and passes it to the original function. The function works for x1<x2 and x1>x2.

Usage

convert.side(f, x1, x2)

Arguments

f a function defined on [x1,x2]
x1 numeric vector of length 1; if longer, only the first element is used
x2 numeric vector of length 1; if longer, only the first element is used
DiscontinuousFuzzyNumber

Value

a new function defined on [0,1] (scaled input)

See Also

FuzzyNumber

Other FuzzyNumber.method: alphacut, alphaInterval, alphaInterval, alphaInterval, alphaInterval, ambiguity, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, core, distance, distance, distance, distance, distance, evaluate, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedValue, Extract, Extract, Extract, Extract, FuzzyNumber-class, integrateAlpha, integrateAlpha, piecewiseLinearApproximation, plot, plot, plot, show, show, show, show, show, supp, trapezoidalApproximation, value, weightedExpectedValue, width

core

Calculate the core of a fuzzy number

Description

We have \( \text{supp}(A) := [a_2, a_3] \).

Methods

signature(object = "FuzzyNumber")

See Also

Other FuzzyNumber.method: alphacut, alphaInterval, alphaInterval, alphaInterval, alphaInterval, ambiguity, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, convert.side, distance, distance, distance, distance, distance, evaluate, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedValue, Extract, Extract, Extract, Extract, Extract, FuzzyNumber-class, integrateAlpha, integrateAlpha, piecewiseLinearApproximation, plot, plot, plot, show, show, show, show, show, supp, trapezoidalApproximation, value, weightedExpectedValue, width

DiscontinuousFuzzyNumber

Creates a fuzzy number with possibly discontinuous side functions or alpha-cut bounds

Description

For convenience, objects of class DiscontinuousFuzzyNumber may be created with this function.
DiscontinuousFuzzyNumber

Usage

DiscontinuousFuzzyNumber(a1, a2, a3, a4,
  lower = function(a) rep(NA_real_, length(a)),
  upper = function(a) rep(NA_real_, length(a)),
  left = function(x) rep(NA_real_, length(x)),
  right = function(x) rep(NA_real_, length(x)),
  discontinuities.left = numeric(0),
  discontinuities.right = numeric(0),
  discontinuities.lower = numeric(0),
  discontinuities.upper = numeric(0))

Arguments

a1 a number specifying left bound of the support
a2 a number specifying left bound of the core
a3 a number specifying right bound of the core
a4 a number specifying right bound of the support
lower lower alpha-cut bound generator; a nondecreasing function [0,1]->[0,1] or returning NA_real_
upper upper alpha-cut bound generator; a nonincreasing function [0,1]->[1,0] or returning NA_real_
left lower side function generator; a nondecreasing function [0,1]->[0,1] or returning NA_real_
right upper side function generator; a nonincreasing function [0,1]->[1,0] or returning NA_real_
 discontinuities.left nondecreasingly sorted numeric vector with elements in (0,1), possibly of length 0
 discontinuities.right nondecreasingly sorted numeric vector with elements in (0,1), possibly of length 0
 discontinuities.lower nondecreasingly sorted numeric vector with elements in (0,1), possibly of length 0
 discontinuities.upper nondecreasingly sorted numeric vector with elements in (0,1), possibly of length 0

Value

Object of class DiscontinuousFuzzyNumber
DiscontinuousFuzzyNumber-class

S4 class representing a fuzzy number with discontinuous side functions or alpha-cut bounds

Description

S4 class representing a fuzzy number with discontinuous side functions or alpha-cut bounds

Slots

- discontinuities.left: Object of class "numeric"
- discontinuities.right: Object of class "numeric"
- discontinuities.lower: Object of class "numeric"
- discontinuities.upper: Object of class "numeric"

Extends

Class FuzzyNumber, directly.

See Also

DiscontinuousFuzzyNumber for a convenient constructor

Examples

showClass("DiscontinuousFuzzyNumber")

distance Calculate the distance between two FNs

Description

This is done by numerical integration

Arguments

- type one of "Euclidean", "EuclideanSquared"
- rel.tol numeric;

Value

the calculated distance
evaluate

Methods

signature(object1 = "FuzzyNumber", object2 = "FuzzyNumber")
signature(object1 = "DiscontinuousFuzzyNumber", object2 = "FuzzyNumber")
signature(object1 = "FuzzyNumber", object2 = "DiscontinuousFuzzyNumber")
signature(object1 = "DiscontinuousFuzzyNumber", object2 = "DiscontinuousFuzzyNumber")

References


See Also

integrate, integrate_discont_val

Other FuzzyNumber.method: alphacut, alphaInterval, alphaInterval, alphaInterval, alphaInterval, ambiguity, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, convert.side, core, evaluate, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, Extract, Extract, Extract, Extract, Extract, FuzzyNumber-class, integrateAlpha, integrateAlpha, piecewiseLinearApproximation, plot, plot, plot, show, show, show, show, supp, trapezoidalApproximation, value, weightedExpectedValue, width

evaluate  Evaluate the membership function

Description

This function returns the value(s) of the membership function of a fuzzy number at given point(s).

Value

Value of the membership function at given points

Methods

signature(object = "FuzzyNumber", alpha = "numeric")

See Also

Other FuzzyNumber.method: alphacut, alphaInterval, alphaInterval, alphaInterval, alphaInterval, ambiguity, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, convert.side, core, distance, distance, distance, distance, distance, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, Extract, Extract, Extract, Extract, Extract, FuzzyNumber-class, integrateAlpha, integrateAlpha, piecewiseLinearApproximation, plot, plot, plot, show, show, show, show, supp, trapezoidalApproximation, value, weightedExpectedValue, width
**Examples**

```r
t <- TrapezoidalFuzzyNumber(1,2,3,4);
print(evaluate(t, seq(0.5,by=0.5)));
```

---

**expectedInterval**

*Calculate the expected interval of a fuzzy number*

### Description

We have $EI(A) := [\int_0^1 A_L(\alpha) \, d\alpha, \int_0^1 A_U(\alpha) \, d\alpha]$, see (Duboid, Prade, 1987).

### Details

Note that this may be done with numeric integration (for instances of the `FuzzyNumber` and `DiscontinuousFuzzyNumber` class).

### Methods

- `signature(object = "FuzzyNumber")` (numerical integration used)
- `signature(object = "TrapezoidalFuzzyNumber")` (exact)
- `signature(object = "PiecewiseLinearFuzzyNumber")` (exact)
- `signature(object = "PowerFuzzyNumber")` (exact)

### References


### See Also

- `integrateAlpha`

Other `FuzzyNumber` methods: `alphacut, alphaInterval, alphaInterval, alphaInterval, alphaInterval, ambiguity, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, convert.side, core, distance, distance, distance, distance, evaluate, expectedValue, Extract, Extract, Extract, Extract, FuzzyNumber-class, integrateAlpha, integrateAlpha, piecewiseLinearApproximation, plot, plot, plot, show, show, show, show, show, supp, trapezoidalApproximation, value, weightedExpectedValue, width`
Calculate the expected value of a fuzzy number (defuzzify)

Description

The expected value of $A$ is defined as $EV_w(A) := EI_U(A) - EI_L(A)$, where $EI$ is the expectedInterval.

Methods

signature(object = "FuzzyNumber")

See Also

expectedInterval on which this function is based, and also weightedExpectedValue

Other FuzzyNumber.method: alphacut, alphaInterval, alphaInterval, alphaInterval, alphaInterval, ambiguity, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, convert, side, core, distance, distance, distance, distance, evaluate, expectedInterval, expectedInterval, expectedInterval, Extract, Extract, Extract, Extract, Extract, Extract, FuzzyNumber-class, integrateAlpha, integrateAlpha, piecewiseLinearApproximation, plot, plot, plot, show, show, show, show, supp, trapezoidalApproximation, value, weightedExpectedValue, width

FuzzyNumber slot accessor (read-only)

Description

For possible slot names see man pages for class FuzzyNumber and its derivatives

Arguments

i slot name

j not used

drop not used

Value

slot value

See Also

Other FuzzyNumber.method: alphacut, alphaInterval, alphaInterval, alphaInterval, alphaInterval, ambiguity, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, convert, side, core, distance, distance, distance, distance, evaluate, expectedInterval, expectedInterval, expectedInterval, extract, Extract, Extract, Extract, Extract, Extract, Extract, FuzzyNumber-class, integrateAlpha, integrateAlpha, piecewiseLinearApproximation, plot, plot, plot, show, show, show, show, show, supp, trapezoidalApproximation, value, weightedExpectedValue, width
Examples

```r
A <- FuzzyNumber(1,2,3,4)
A['a1']
A['right']
```

**FuzzyNumber**  
*Creates a Fuzzy Number*

**Description**

For convenience, objects of class FuzzyNumber (see **FuzzyNumber-class**) may be created with this function.

**Usage**

```r
FuzzyNumber(a1, a2, a3, a4,
    lower = function(a) rep(NA_real_, length(a)),
    upper = function(a) rep(NA_real_, length(a)),
    left = function(x) rep(NA_real_, length(x)),
    right = function(x) rep(NA_real_, length(x)))
```

**Arguments**

- **a1**: a number specifying left bound of the support
- **a2**: a number specifying left bound of the core
- **a3**: a number specifying right bound of the core
- **a4**: a number specifying right bound of the support
- **lower**: lower alpha-cut bound generator; a nondecreasing function \([0,1] \rightarrow [0,1]\) or returning `NA_real_`
- **upper**: upper alpha-cut bound generator; a nonincreasing function \([0,1] \rightarrow [1,0]\) or returning `NA_real_`
- **left**: lower side function generator; a nondecreasing function \([0,1] \rightarrow [0,1]\) or returning `NA_real_`
- **right**: upper side function generator; a nonincreasing function \([0,1] \rightarrow [1,0]\) or returning `NA_real_`

**Value**

Object of class FuzzyNumber
FuzzyNumber-class

S4 class representing a fuzzy number

**Description**

Formally, a fuzzy number $A$ (Dubois, Prade, 1978) is a fuzzy subset of the real line $\mathbb{R}$ with membership function $\mu$ given by:
where \( a_1, a_2, a_3, a_4 \in R, a_1 \leq a_2 \leq a_3 \leq a_4 \), \( \text{left} : [0, 1] \rightarrow [0, 1] \) is a nondecreasing function called the left side generator of \( A \), and \( \text{right} : [0, 1] \rightarrow [0, 1] \) is a nonincreasing function called the right side generator of \( A \).

Alternatively, it may be shown that each fuzzy number \( A \) may be uniquely determined by specifying its \( \alpha \)-cuts, \( A(\alpha) \). We have \( A(0) = [a_1, a_4] \) and

\[
A(\alpha) = [a_1 + (a_2 - a_1) \times \text{lower}(\alpha), a_3 + (a_4 - a_3) \times \text{upper}(\alpha)]
\]

for \( 0 < \alpha \leq 1 \), where \( \text{lower} : [0, 1] \rightarrow [0, 1] \) and \( \text{upper} : [0, 1] \rightarrow [0, 1] \) are, respectively, strictly increasing and decreasing functions satisfying \( \text{lower}(\alpha) = \inf \{ x : \mu(x) \geq \alpha \} \) and \( \text{upper}(\alpha) = \sup \{ x : \mu(x) \geq \alpha \} \).

Please note that many algorithms that deal with fuzzy numbers often use \( \alpha \)-cuts rather than side functions.

**Details**

Note that the **FuzzyNumbers** package also deals with particular types of fuzzy numbers like trapezoidal, piecewise linear, or “parametric” FNs (see **TrapezoidalFuzzyNumber-class PiecewiseLinearFuzzyNumber-class PowerFuzzyNumber-class, DiscontinuousFuzzyNumber-class**)

**Slots**

- **a1**: Single numeric value specifying the left bound for the support.
- **a2**: Single numeric value specifying the left bound for the core.
- **a3**: Single numeric value specifying the right bound for the core.
- **a4**: Single numeric value specifying the right bound for the support.
- **lower**: A nondecreasing function \([0,1]\rightarrow[0,1]\) that gives the lower alpha-cut bound.
- **upper**: A nonincreasing function \([0,1]\rightarrow[1,0]\) that gives the upper alpha-cut bound.
- **left**: A nondecreasing function \([0,1]\rightarrow[0,1]\) that gives the left side function.
- **right**: A nonincreasing function \([0,1]\rightarrow[1,0]\) that gives the right side function.

**See Also**

- **FuzzyNumber** for a convenient constructor, **convert.side** for creating side functions generators, **convert.alpha** for creating alpha-cut bounds generators, **approx.invert** for inverting side functions/alpha-cuts numerically.

**integrateAlpha**

Examples

```r
showClass("FuzzyNumber")
showMethods(classes="FuzzyNumber")
```

---

**integrateAlpha**

Numerically integrate a transformed or weighted lower or upper alpha-cut bound of a fuzzy number

---

**Description**

Numerically integrate a transformed or weighted lower or upper alpha-cut bound of a fuzzy number

**Arguments**

- `weight` a function or NULL
- `transform` a function or NULL
- `rel.tol` numeric

**Methods**

- `signature(object = "FuzzyNumber", which="character", from="numeric", to="numeric")`
- `signature(object = "DiscontinuousFuzzyNumber", which="character", from="numeric", to="numeric")`

**See Also**

`integrate_discont_val`

Other FuzzyNumber method: `alphacut, alphaInterval, alphaInterval, alphaInterval, alphaInterval, ambiguity, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, convert.side, core, distance, distance, distance, distance, evaluate, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedValue, Extract, Extract, Extract, Extract, FuzzyNumber-class, piecewiseLinearApproximation, plot, plot, plot, show, show, show, show, show, supp, trapezoidalApproximation, value, weightedExpectedValue, width`
integrate_discont_val  Integrate a function with at most finite number of discontinuities

Description

The function uses multiple calls to \texttt{integrate}.

Usage

\begin{verbatim}
integrate_discont_val(f, from, to,
  discontinuities = numeric(0),
  rel.tol = .Machine$double.eps^0.35, ...)
\end{verbatim}

Arguments

- \texttt{f}: an R function taking a numeric vector of length 1 as its first argument and returning a numeric vector of length 1
- \texttt{from}: the lower limit of integration
- \texttt{to}: the upper limit of integration
- \texttt{discontinuities}: nondecreasingly sorted numeric vector which indicates the points at which \texttt{f} is discontinuous
- \texttt{rel.tol}: relative accuracy requested
- \texttt{...}: further arguments to be passed to the \texttt{integrate} function.

Value

the estimate of the integral

piecewiseLinearApproximation  Piecewise linear approximation of a fuzzy number

Description

This method finds a piecewise linear approximation $P(A)$ of a given fuzzy number $A$ by using the algorithm specified by the \texttt{method} parameter.

Arguments

- \texttt{method}: one of: "NearestEuclidean", "ApproximateNearestEuclidean", "Naive"
- \texttt{verbose}: logical
- \texttt{...}: further arguments passed to \texttt{integrateAlpha}
- \texttt{knot.n}: number of knots
- \texttt{knot.alpha}: alpha-cuts for knots
- \texttt{optim.control}: a list of control params for \texttt{optim}
Details

method may be one of:

1. Naive: We have core(A)==core(T(A)) and supp(A)==supp(T(A)) and the knots are taken
directly from the specified alpha cuts (linear interpolation).
2. NearestEuclidean: see (Coroianu, Gagolewski, Grzegorzewski, 2013), only for knot.n==1;
uses numerical integration, see integrateAlpha
3. ApproximateNearestEuclidean: this is done via numeric optimization ("Nelder-Mead" al-
gorithm); uses numerical integration, see integrateAlpha

References

Coroianu L., Gagolewski M., Grzegorzewski P. (2013), Nearest Piecewise Linear Approximation

See Also
trapezoidalApproximation

Other FuzzyNumber.method: alphacut, alphaInterval, alphaInterval, alphaInterval, alphaInterval,
ambiguity, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, convert.side,
core, distance, distance, distance, distance, evaluate, expectedInterval, expectedInterval,
expectedInterval, expectedInterval, expectedValue, Extract, Extract, Extract, Extract,
FuzzyNumber-class, integrateAlpha, integrateAlpha, plot, plot, plot, show, show, show,
show, supp, trapezoidalApproximation, value, weightedExpectedValue, width

Examples

(A <- FuzzyNumber(-1,0,1,3,lower=function(x) sqrt(x),upper=function(x) 1-sqrt(x)))
(PA <- piecewiseLinearApproximation(A, "NearestEuclidean", knot.n=1, knot.alpha=0.2))

PiecewiseLinearFuzzyNumber

Creates a piecewise linear fuzzy number

Description

For convenience, objects of class PiecewiseLinearFuzzyNumber may be created with this func-
tion.

Usage

PiecewiseLinearFuzzyNumber(a1, a2, a3, a4, knot.n = 0,
knot.alpha = numeric(0), knot.left = numeric(0),
knot.right = numeric(0))
Arguments

- a1: a number specifying left bound of the support
- a2: a number specifying left bound of the core
- a3: a number specifying right bound of the core
- a4: a number specifying right bound of the support
- knot.n: the number of knots
- knot.alpha: knot.n alpha-cut values at knots
- knot.left: knot.n knots on the left side; a nondecreasingly sorted vector with elements in [a1,a2]
- knot.right: knot.n knots on the right side; a nondecreasingly sorted vector with elements in [a3,a4]

Value

Object of class `PiecewiseLinearFuzzyNumber`

Description

A piecewise linear fuzzy number (PLFN) has side functions and alpha-cut bounds that linearly interpolate a given set of points (at fixed alpha-cuts).

Slots

- knot.n: Object of class "numeric"
- knot.alpha: Object of class "numeric"
- knot.left: Object of class "numeric"
- knot.right: Object of class "numeric"

Extends

Class `FuzzyNumber`, directly.

See Also

`PiecewiseLinearFuzzyNumber` for a convenient constructor

Examples

`showClass("PiecewiseLinearFuzzyNumber")`
Plot a fuzzy number

Description

Note that if from > a1 then it is set to a1.

Arguments

from numeric;
to numeric;
n numeric;
at.alpha numeric vector;
draw.membership.function logical;
draw.alphacuts defaults !draw.membership.function
xlab character;
ylab character;
xlim numeric;
ylim numeric;
type character; defaults "l"
col see plot.default
lty see plot.default
pch see plot.default
lwd see plot.default
shadowintensity for shadowed sets;
shadowangle for shadowed sets;
shadowcol for shadowed sets;
shadowborder for shadowed sets;
add logical;
... further arguments passed to plot.default

Methods

signature(x = "FuzzyNumber", y = "missing")
signature(x = "TrapezoidalFuzzyNumber", y = "missing")
signature(x = "PiecewiseLinearFuzzyNumber", y = "missing")
See Also

Other FuzzyNumber.method: alphacut, alphaInterval, alphaInterval, alphaInterval, alphaInterval, ambiguity, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, convert.side, core, distance, distance, distance, distance, evaluate, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, Extract, Extract, Extract, Extract, FuzzyNumber-class, integrateAlpha, integrateAlpha, piecewiseLinearApproximation, show, show, show, show, supp, trapezoidalApproximation, value, weightedExpectedValue, width

Examples

plot(FuzzyNumber(0,1,2,3), col="gray")
plot(FuzzyNumber(0,1,2,3, left=function(x) x^2, right=function(x) 1-x^3), add=TRUE)
plot(FuzzyNumber(0,1,2,3, lower=function(x) x, upper=function(x) 1-x), add=TRUE, col=2)

---

PowerFuzzyNumber

Creates a "parametric" fuzzy number with sides given by power functions

Description

For convenience, objects of class PowerFuzzyNumber may be created with this function.

Usage

PowerFuzzyNumber(a1, a2, a3, a4, p.left = 1, p.right = 1)

Arguments

- a1: a number specifying left bound of the support
- a2: a number specifying left bound of the core
- a3: a number specifying right bound of the core
- a4: a number specifying right bound of the support
- p.left: a positive number specifying the exponent for the left side
- p.right: a positive number specifying the exponent for the right side

Value

Object of class PowerFuzzyNumber
PowerFuzzyNumber-class

S4 class representing a fuzzy number with sides given by power functions

Description

S4 class representing a fuzzy number with sides given by power functions

Slots

p.left: Object of class "numeric"; 1.0 to get a trapezoidal FN
p.right: Object of class "numeric"; 1.0 to get a trapezoidal FN

Extends

Class FuzzyNumber, directly.

See Also

PowerFuzzyNumber for a convenient constructor

Examples

showClass("PowerFuzzyNumber")

Description

Print basic information on a fuzzy number

Methods

signature(object = "FuzzyNumber")
signature(object = "TrapezoidalFuzzyNumber")
signature(object = "PiecewiseLinearFuzzyNumber")
signature(object = "PowerFuzzyNumber")
trapezoidalApproximation

TRapezoidal approximation of a fuzzy number

Description
This method finds a trapezoidal approximation \( T(A) \) of a given fuzzy number \( A \) by using the algorithm specified by the method parameter.

Arguments

- \textbf{method}\:
  one of: "NearestEuclidean", "ExpectedIntervalPreserving", "SupportCoreRestricted", "Naive"
- \textbf{verbose}\:
  logical
- \textbf{...}\:
  further arguments passed to \texttt{integrateAlpha}
Details

method may be one of:

1. Naive: We have \( \text{core}(A) = \text{core}(T(A)) \) and \( \text{supp}(A) = \text{supp}(T(A)) \)

2. ExpectedIntervalPreserving: L2-nearest trapezoidal approximation preserving the expected interval given in (Grzegorzewski, 2010; Ban, 2008; Yeh, 2008) Unfortunately, for highly skewed membership functions this approximation operator may have quite unfavourable behavior. E.g. if \( \text{Val}(A) < \text{EV}_{1/3}(A) \) or \( \text{Val}(A) > \text{EV}_{2/3}(A) \), then it may happen that the core of the output and the core of the original fuzzy number \( A \) are disjoint (cf. Grzegorzewski, Pasternak-Winiarska, 2011)

3. SupportCoreRestricted: This method was proposed in (Grzegorzewski, Pasternak-Winiarska, 2011). L2-nearest trapezoidal approximation with constraints \( \text{core}(A) \subseteq \text{core}(T(A)) \) and \( \text{supp}(T(A)) \subseteq \text{supp}(A) \), i.e. for which each point that surely belongs to \( A \) also belongs to \( T(A) \), and each point that surely does not belong to \( A \) also does not belong to \( T(A) \).

4. NearestEuclidean: see (Ban, 2009); uses numerical integration, see \texttt{integrateAlpha}

References


See Also

\texttt{piecewiseLinearApproximation}

Other FuzzyNumber method: \texttt{alphacut, alphaInterval, alphaInterval, alphaInterval, alphaInterval, ambiguity, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, convert.side, core, distance, distance, distance, distance, distance, distance, evaluate, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedValue, Extract, Extract, Extract, Extract, FuzzyNumber-class, integrateAlpha, integrateAlpha, piecewiseLinearApproximation, plot, plot, plot, plot, show, show, show, show, show, supp, value, weightedExpectedValue, width}

Examples

\begin{verbatim}
(A <- FuzzyNumber(-1,0,1,4,lower=function(x) sqrt(x),upper=function(x) 1-sqrt(x)))
(TA <- trapezoidalApproximation(A, "ExpectedIntervalPreserving") # Note that cores are disjoint!
expectedInterval(A)
expectedInterval(TA)
\end{verbatim}
TrapezoidalFuzzyNumber

*Creates a trapezoidal fuzzy number*

---

**Description**

For convenience, objects of class `TrapezoidalFuzzyNumber` may be created with this function.

**Usage**

```r
TrapezoidalFuzzyNumber(a1, a2, a3, a4)
```

**Arguments**

- `a1`: a number specifying left bound of the support
- `a2`: a number specifying left bound of the core
- `a3`: a number specifying right bound of the core
- `a4`: a number specifying right bound of the support

**Value**

Object of class `TrapezoidalFuzzyNumber`

---

**TrapezoidalFuzzyNumber-class**

*S4 class representing a trapezoidal fuzzy number*

---

**Description**

S4 class representing a trapezoidal fuzzy number

**Extends**

Class `FuzzyNumber`, directly.

**See Also**

`TrapezoidalFuzzyNumber` for a convenient constructor

**Examples**

```r
showClass("DiscontinuousFuzzyNumber")
```
### value

**Calculate the value of a fuzzy number (defuzzify)**

**Description**

The value of $A$ (Delgado et al, 1998) is defined as $\text{val}(A) := \int_0^1 \alpha (A_L(\alpha) + A_U(\alpha)) \, d\alpha$.

**Methods**

signature(object = "FuzzyNumber")

**References**


**See Also**

* alphaInterval on which this function is based

Other FuzzyNumber.method: alphacut, alphaInterval, alphaInterval, alphaInterval, alphaInterval, ambiguity, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, convert.side, core, distance, distance, distance, distance, evaluate, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedValue, Extract, Extract, Extract,Extract, FuzzyNumber-class, integrateAlpha, integrateAlpha, piecewiseLinearApproximation, plot, plot, plot, show, show, show, show, supp, trapezoidalApproximation, weightedExpectedValue, width

### weightedExpectedValue

**Calculate the weighted expected value of a fuzzy number**

**Description**

The weighted $(0 < w < 1)$ expected value of $A$ is defined as $\text{EV}_w(A) := (1 - w)EI_U(A) - wEI_L(A)$, where $EI$ is the expectedInterval.

**Methods**

signature(object = "FuzzyNumber", w = "numeric")
See Also

expectedInterval on which this function is based

Other FuzzyNumber.method: alphasum, alphaInterval, alphaInterval, alphaInterval, alphaInterval, ambiguity, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, convert.side, core, distance, distance, distance, distance, distance, distance, distance, distance, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, Extract, Extract, Extract, Extract, Extract, FuzzyNumber-class, integrateAlpha, integrateAlpha, piecewiseLinearApproximation, plot, plot, plot, show, show, show, show, show, show, show, supp, trapezoidalApproximation, value, weightedExpectedValue

---

width

**Calculate the width of a fuzzy number**

Description

The width is a measure of nonspecificity of a fuzzy number.

Details

The width of $A$ is defined as $width(A) := EI_U(A) - EI_L(A)$, where $EI$ is the expectedInterval.

Methods

signature(object = "FuzzyNumber")

References


See Also

expectedInterval on which this function is based

Other FuzzyNumber.method: alphasum, alphaInterval, alphaInterval, alphaInterval, alphaInterval, ambiguity, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, Arithmetic, convert.side, core, distance, distance, distance, distance, distance, distance, distance, evaluate, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, expectedInterval, Extract, Extract, Extract, Extract, Extract, FuzzyNumber-class, integrateAlpha, integrateAlpha, piecewiseLinearApproximation, plot, plot, plot, show, show, show, show, show, supp, trapezoidalApproximation, value, weightedExpectedValue
Index

*,PiecewiseLinearFuzzyNumber, numeric-method (Arithmetic), 7
*,TrapezoidalFuzzyNumber, numeric-method (Arithmetic), 7
*,numeric,FuzzyNumber-method (Arithmetic), 7
+,TrapezoidalFuzzyNumber,TrapezoidalFuzzyNumber-method (Arithmetic), 7
-,FuzzyNumber,ANY-method (Arithmetic), 7
-,TrapezoidalFuzzyNumber,TrapezoidalFuzzyNumber-method (Arithmetic), 7
[,DiscontinuousFuzzyNumber-method (Extract), 15
[,FuzzyNumber-method (Extract), 15
[,PiecewiseLinearFuzzyNumber-method (Extract), 15
[,PowerFuzzyNumber-method (Extract), 15
alphacut, 4, 5, 7, 10, 13–15, 18, 19, 21, 24, 26, 27, 29, 30
alphacut,FuzzyNumber,numeric-method (alphacut), 4
alphaInterval, 5, 6, 7, 10, 13–15, 18, 19, 21, 24, 26, 27, 29, 30
alphaInterval,FuzzyNumber-method (alphaInterval), 5
alphaInterval,PiecewiseLinearFuzzyNumber-method (alphaInterval), 5
alphaInterval,PowerFuzzyNumber-method (alphaInterval), 5
alphaInterval,TrapezoidalFuzzyNumber-method (alphaInterval), 5
ambiguity, 5, 6, 7, 10, 13–15, 18, 19, 21, 24, 26, 27, 29, 30
ambiguity,FuzzyNumber-method (ambiguity), 6
approx.invert, 6, 18
Arithmetic, 5, 6, 7, 10, 13–15, 18, 19, 21, 24, 26, 27, 29, 30
as.FuzzyNumber, 8
as.PiecewiseLinearFuzzyNumber, 8
convert.alpha, 9, 18
convert.side, 5–7, 9, 10, 13–15, 18, 19, 21, 24, 26, 27, 29, 30
core, 5–7, 10, 10, 13–15, 18, 19, 21, 24, 26, 29, 30
core,FuzzyNumber-method (core), 10
DiscontinuousFuzzyNumber, 3, 10, 10, 11, 12
DiscontinuousFuzzyNumber-class, 12
distance, 5–7, 10, 12, 13–15, 18, 19, 21, 24, 26, 27, 29, 30
distance,DiscontinuousFuzzyNumber,DiscontinuousFuzzyNumber-method (distance), 12
distance,DiscontinuousFuzzyNumber,FuzzyNumber-method (distance), 12
distance,FuzzyNumber,DiscontinuousFuzzyNumber-method (distance), 12
distance,FuzzyNumber,FuzzyNumber-method (distance), 12
evaluate, 5–7, 10, 13, 14, 15, 18, 19, 21, 24, 26, 27, 29, 30
evaluate,FuzzyNumber,numeric-method (evaluate), 13
expectedInterval, 5–7, 10, 13, 14, 15, 18, 19, 21, 24, 26, 27, 29, 30
expectedInterval,FuzzyNumber-method (expectedInterval), 14
expectedInterval,PowerFuzzyNumber-method (expectedInterval), 14
expectedInterval,PiecewiseLinearFuzzyNumber-method (expectedInterval), 14
expectedInterval,TrapezoidalFuzzyNumber-method (expectedInterval), 14
expectedValue, 3, 5–7, 10, 13–15, 18, 19, 21, 24, 26, 27, 29, 30
expectedValue, FuzzyNumber-method
(expectedValue), 15
Extract, 5–7, 10, 13–15, 15, 18, 19, 21, 24,
26, 27, 29, 30
FuzzyNumber, 3, 7, 9, 10, 12, 16, 18, 22, 25, 28
FuzzyNumber-class, 17
FuzzyNumbers-package, 3
integrate, 13, 20
integrate_discont_val, 13, 19, 20
integrateAlpha, 5–7, 10, 13–15, 18, 19, 20,
21, 24, 26, 27, 29, 30
integrateAlpha, DiscontinuousFuzzyNumber, character,
numeric, numeric-method (integrateAlpha), 19
integrateAlpha, FuzzyNumber, character, numeric,
numeric-method (integrateAlpha), 19
optim, 20
piecewiseLinearApproximation, 3, 5–7, 10,
13–15, 18, 19, 20, 24, 26, 27, 29, 30
piecewiseLinearApproximation, FuzzyNumber-method
(piecewiseLinearApproximation), 20
PiecewiseLinearFuzzyNumber, 3, 9, 21, 21,
22
PiecewiseLinearFuzzyNumber-class, 22
plot, 3, 5–7, 10, 13–15, 18, 19, 21, 23, 26, 27,
29, 30
plot, FuzzyNumber, missing-method (plot), 23
plot, PiecewiseLinearFuzzyNumber, missing-method
(plot), 23
plot, TrapezoidalFuzzyNumber, missing-method
(plot), 23
plot.default, 23
PowerFuzzyNumber, 3, 24, 25
PowerFuzzyNumber-class, 25
show, 5–7, 10, 13–15, 18, 19, 21, 24, 25, 26,
27, 29, 30
show, FuzzyNumber-method (show), 25
show, PiecewiseLinearFuzzyNumber-method
(show), 25
show, PowerFuzzyNumber-method (show), 25
show, TrapezoidalFuzzyNumber-method
(show), 25
supp, 5–7, 10, 13–15, 18, 19, 21, 24, 26, 26,
27, 29, 30
supp, FuzzyNumber-method (supp), 26
trapezoidalApproximation, 3, 5–7, 10,
13–15, 18, 19, 21, 24, 26, 26, 29, 30
trapezoidalApproximation, FuzzyNumber-method
(trapezoidalApproximation), 26
TrapezoidalFuzzyNumber, 3, 28, 28
TrapezoidalFuzzyNumber-class, 28
value, 3, 5–7, 10, 13–15, 18, 19, 21, 24, 26,
27, 29, 30
value, FuzzyNumber-method (value), 29
weightedExpectedValue, 3, 13–15, 18,
19, 21, 24, 26, 27, 29, 30
weightedExpectedValue, FuzzyNumber, numeric-method
(weightedExpectedValue), 29
width, 5–7, 10, 13–15, 18, 19, 21, 24, 26, 27,
29, 30, 30
width, FuzzyNumber-method (width), 30