API Documentation

API Documentation

April 10, 2017

Contents

1 Package fuzzycreator

1.1 Modules

1.2 Variables

Name	Description
___package___	**Value:** `None`

2 Module fuzzycreator.fuzzy_exceptions

This module lists fuzzy set based exceptions.

These are regarding non-existent parts of fuzzy sets, e.g. empty alpha-cuts, empty zLevels.

2.1 Variables

Name	Description
___package___	**Value:** `None`

2.2 Class AlphaCutError

object
exceptions.BaseException
exceptions.Exception
fuzzycreator.fuzzy_exceptions.AlphaCutError

The alpha-cut exceeds the height of the fuzzy set.

2.2.1 Methods

Inherited from exceptions.Exception

___init___(), ___new___()

Inherited from exceptions.BaseException

___delattr___(), ___getattribute___(), ___getitem___(), ___getslice___(), ___reduce___(), ___repr___(), ___setattr___(), ___setstate___(), ___str___(), ___unicode___()

Inherited from object

___format___(), ___hash___(), ___reduce_ex___(), ___sizeof___(), ___subclasshook___()

2.2.2 Properties

Name	Description
Inherited from exceptions.BaseException	
args, message	
Inherited from object	

continued on next page

Name	Description
___class___	

2.3 Class ZLevelError

object
exceptions.BaseException
exceptions.Exception
fuzzycreator.fuzzy_exceptions.ZLevelError

The zlevel exceeds the secondary height of the fuzzy set.

2.3.1 Methods

Inherited from exceptions.Exception

___init___(), ___new___()

Inherited from exceptions.BaseException

___delattr___(), ___getattribute___(), ___getitem___(), ___getslice___(), ___reduce___(), ___repr___(), ___setattr___(), ___setstate___(), ___str___(), ___unicode___()

Inherited from object

___format___(), ___hash___(), ___reduce_ex___(), ___sizeof___(), ___subclasshook___()

2.3.2 Properties

Name	Description
Inherited from exceptions.BaseException args, message	
Inherited from object ___class___	

3 Package fuzzycreator.fuzzy_sets

3.1 Modules

- **discrete_t1_fuzzy_set**: This module is used to create a discrete type-1 fuzzy set.
(Section 4, p. 9)
- **discrete_t2_fuzzy_set**: This module is used to create a discrete type-2 fuzzy set.
(Section 5, p. 10)
- **fuzzy_set**: This module is used to create a type-1 fuzzy set.
(Section 6, p. 12)
- **general_t2_fuzzy_set**: This module is used to create a general type-2 fuzzy set.
(Section 7, p. 14)
- **interval_t2_fuzzy_set**: This module is used to create an interval type-2 fuzzy set.
(Section 8, p. 16)
- **polling_t1_fuzzy_set**: This module is used to create a continuous version of a type-1 set.
(Section 9, p. 18)
- **t2_aggregated_fuzzy_set**: This module is for aggregating type-1 fuzzy sets into a type-2 fuzzy set.
(Section 10, p. 19)

3.2 Variables

Name	Description
___package___	**Value:** `None`

4 Module fuzzycreator.fuzzy_sets.discrete_t1_fuzzy_set

This module is used to create a discrete type-1 fuzzy set.

4.1 Variables

Name	Description
___package___	**Value:** `'fuzzycreator.fuzzy_sets'`

4.2 Class DiscreteT1FuzzySet

Create a discrete type-1 fuzzy set.

4.2.1 Methods

___init___(*self*, *points*)

Create a discrete type-1 fuzzy set using a dict of x,mu pairs.

calculate_membership(*self*, *x*)

Calculate the membership of x within the uod.

Returns a Decimal value.

calculate_alpha_cut(*self*, *alpha*)

Calculate the alpha-cut of the function within the uod.

alpha must be greater than 0 and less than the function height. Returns a two-tuple.

calculate_centroid(*self*)

Calculate the centroid x-value of the fuzzy set.

plot_set(*self*, *filename*=`None`)

Plot a graph of the fuzzy set.

If filename is None, the plot is displayed. If a filename is given, the plot is saved to the given location.

5 Module fuzzycreator.fuzzy_sets.discrete_t2_fuzzy_set

This module is used to create a discrete type-2 fuzzy set.

5.1 Variables

Name	Description
___package___	**Value:** `'fuzzycreator.fuzzy_sets'`

5.2 Class DiscreteT2FuzzySet

Create a discrete type-2 fuzzy set.

5.2.1 Methods

___init___(*self*, *points*)

Create a discrete type-1 fuzzy set using a dict as {x: {mu: z}}.

validate_zlevel(*self*, *z*)

Find the closest valid zlevel.

Checks if the zlevel at z exists. If it exists then return z. If not, then return the closest zlevel that encompasses z.

calculate_membership(*self*, *x*, *z*)

Calculate the primary membership of x at the zlevel z.

calculate_secondary_membership(*self*, *x*, *mu*)

Calculate the secondary membership of x at primary membership y.

calculate_alpha_cut_lower(*self*, *alpha*, *z*=`0`)

Calculate the alpha-cut of the lower membership function.

alpha must be greater than 0 and less than the function height. Returns a two-tuple.

calculate_alpha_cut_upper(*self*, *alpha*, *z*=0)

Calculate the alpha-cut of the lower membership function.

alpha must be greater than 0 and less than the function height. Returns a two-tuple.

6 Module fuzzycreator.fuzzy_sets.fuzzy_set

This module is used to create a type-1 fuzzy set.

6.1 Variables

Name	Description
___package___	**Value:** `'fuzzycreator.fuzzy_sets'`

6.2 Class FuzzySet

Create a type-1 fuzzy set.

6.2.1 Methods

___init___(*self*, *membership_function*, *uod*=`None`)

Create a type-1 fuzzy set.

membership_function: a membership function object. uod: the universe of discourse indicated by a two-tuple.

calculate_membership(*self*, *x*)

Calculate the membership of x within the uod.

Returns a Decimal value.

calculate_alpha_cut(*self*, *alpha*)

Calculate the alpha-cut of the function within the uod.

alpha must be greater than 0 and less than the function height. Returns a two-tuple.

calculate_centroid(*self*)

Calculate the centroid x-value of the fuzzy set.

plot_set(*self*, *filename*=`None`)

Plot a graph of the fuzzy set.

If filename is None, the plot is displayed. If a filename is given, the plot is saved to the given location.

7 Module fuzzycreator.fuzzy_sets.general_t2_fuzzy_set

This module is used to create a general type-2 fuzzy set.

7.1 Variables

Name	Description
___package___	**Value:** `'fuzzycreator.fuzzy_sets'`

7.2 Class GeneralT2FuzzySet

Create a zSlices (alpha-plane) based general type-2 fuzzy set.

7.2.1 Methods

___init___(*self*, *mf1*, *mf2*, *zlevels_total*=`None`, *uod*=`None`)

Create a general type-2 fuzzy set.

mf1: first membership function object of lowest zslice mf2: second membership function object of lowest zslice zlevels_total: total number of zlevels uod: the universe of discourse indicated by a two-tuple. Note, the lower and upper membership functions may be assiged in any order to mf1 and mf2.

validate_zlevel(*self*, *z*)

Find the closest valid zlevel.

Checks if the zlevel at z exists. If it exists then return z. If not, then return the closest zlevel that encompasses z.

calculate_membership(*self*, *x*, *z*)

Calculate the primary membership of x at the zlevel z.

calculate_secondary_membership(*self*, *x*, *y*)

Calculate the secondary membership of x at primary membership y.

calculate_alpha_cut_lower(*self*, *x*, *z*)

Calculate the alpha-cut of the lower membership function at z.

alpha must be greater than 0 and less than the function height. Returns a two-tuple.

calculate_alpha_cut_upper(*self*, *x*, *z*)

Calculate the alpha-cut of the upper membership function at z.

alpha must be greater than 0 and less than the function height. Returns a two-tuple.

calculate_centre_of_sets(*self*)

Calculate centre-of-sets type reduction.

Uses the Karnik Mendel algorithm. Returns a two-tuple indicating the boundaries of the type-reduced set.

calculate_overall_centre_of_sets(*self*)

Calculate centre-of-sets type reduction.

Returns the centroid of the centre-of-sets type reduced result.

plot_set(*self*, *filename*=`None`)

Plot a graph of the fuzzy set.

If filename is None, the plot is displayed. If a filename is given, the plot is saved to the given location.

plot_set_3d(*self*)

Plot a 3-dimensional graph of the fuzzy set.

8 Module fuzzycreator.fuzzy_sets.interval_t2_fuzzy_set

This module is used to create an interval type-2 fuzzy set.

8.1 Variables

Name	Description
___package___	**Value:** `'fuzzycreator.fuzzy_sets'`

8.2 Class IntervalT2FuzzySet

Create an interval type-2 fuzzy set.

8.2.1 Methods

___init___(*self*, *mf1*, *mf2*, *uod*=`None`)

Create an interval type-2 fuzzy set.

mf1: first membership function object mf2: second membership function object uod: the universe of discourse indicated by a two-tuple. Note, the lower and upper membership functions may be assiged in any order to mf1 and mf2.

calculate_membership(*self*, *x*)

Calculate the membership of x within the uod.

Returns a two-tuple (lower, upper) of Decimal values.

calculate_alpha_cut_lower(*self*, *alpha*)

Calculate the alpha-cut of the lower membership function.

alpha must be greater than 0 and less than the function height. Returns a two-tuple.

calculate_alpha_cut_upper(*self*, *alpha*)

Calculate the alpha-cut of the upper membership function.

alpha must be greater than 0 and less than the function height. Returns a two-tuple.

plot_set(*self*, *filename*)

Plot a graph of the fuzzy set.

If filename is None, the plot is displayed. If a filename is given, the plot is saved to the given location.

calculate_centre_of_sets(*self*)

Calculate centre-of-sets type reduction.

Uses the Karnik Mendel algorithm. Returns a dict of two-tuples {z:(l, r)} indicating the boundaries of the type-reduced set at each zlevel.

calculate_overall_centre_of_sets(*self*)

Calculate centre-of-sets type reduction.

Returns the centroid of the centre-of-sets type reduced result.

9 Module fuzzycreator.fuzzy_sets.polling_t1_fuzzy_set

This module is used to create a continuous version of a type-1 set.

9.1 Variables

Name	Description
___package___	**Value:** `'fuzzycreator.fuzzy_sets'`

9.2 Class PollingT1FuzzySet

Create a type-1 fuzzy set using the polling technique.

9.2.1 Methods

___init___(*self*, *points*, *uod*=`None`)

Create a discrete type-1 fuzzy set using a dict of x,mu pairs.

calculate_membership(*self*, *x*)

Calculate the membership of x within the uod.

If x is not in self.points but exists between known x-values then linear interpolation is used to calculate its membership. Returns a Decimal value.

calculate_alpha_cut(*self*, *alpha*)

Calculate the alpha-cut of the function within the uod.

alpha must be greater than 0 and less than the function height. Returns a two-tuple.

calculate_centroid(*self*)

Calculate the centroid x-value of the fuzzy set.

plot_set(*self*, *filename*=`None`)

Plot a graph of the fuzzy set.

If filename is None, the plot is displayed. If a filename is given, the plot is saved to the given location.

10 Module fuzzycreator.fuzzy_sets.t2_aggregated_fuzzy_set

This module is for aggregating type-1 fuzzy sets into a type-2 fuzzy set.

The method of aggregation is the same as that used by the interval aggrement approach, details of which are within C. Wagner, S. Miller, J. M. Garibaldi, D. T. Anderson and T. C. Havens, "From Interval-Valued Data to General Type-2 Fuzzy Sets," in IEEE Transactions on Fuzzy Systems, vol. 23, no. 2, pp. 248-269, April 2015. doi: 10.1109/TFUZZ.2014.2310734

10.1 Variables

Name	Description
___package___	**Value:** `'fuzzycreator.fuzzy_sets'`

10.2 Class T2AggregatedFuzzySet

This class is for type-2 Interval Agreement Approach fuzzy sets.

10.2.1 Methods

___init___(*self*, *uod*=`None`)

Initate a type-2 interval agreement approach fuzzy set.

add_membership_function(*self*, *mf*)

Add a type-1 membership function to the fuzzy set.

validate_zlevel(*self*, *z*)

Find the closest valid zlevel.

Checks if the zlevel at z exists. If it exists then return z. If not, then return the closest zlevel that encompasses z.

calculate_membership(*self*, *x*, *z*=`None`)

Calculate the primary membership of x at the zlevel z.

For an interval type-2 fuzzy set, leave z as None.

calculate_secondary_membership(*self*, *x*, *y*)

Calculate the secondary membership value for the given x and y.

calculate_alpha_cut_upper(*self*, *alpha*, *z*=`None`)

Calculate the alpha-cut of the lower membership function at z.

alpha must be greater than 0 and less than the function height. For an interval type-2 fuzzy set, leave z as None. Returns a list containing two-tuples (a list of cuts is always given as any alpha-cut may be non-convex)

calculate_alpha_cut_lower(*self*, *alpha*, *z*)

Calculate the alpha-cut of the lower membership function at z.

alpha must be greater than 0 and less than the function height.

calculate_centroid(*self*)

Calculate the centroid of the fuzzy set.

Calculate the centroid of each zslice and take the weighted average.

plot_set(*self*, *filename*=`None`)

Plot a graph of the fuzzy set.

If filename is None, the plot is displayed. If a filename is given, the plot is saved to the given location.

plot_set_3d(*self*)

Plot a graph of the fuzzy set.

11 Module fuzzycreator.generate_fuzzy_sets

This module is used to generate fuzzy sets from data.

11.1 Functions

generate_gaussian_t1_fuzzy_set(*data*)

Create a Gaussian distributed type-1 fuzzy set from the given data.

generate_gaussian_t2_fuzzy_set(*data*)

Create a Gaussian distributed type-1 fuzzy set from the given data.

generate_discrete_t1_fuzzy_set(*data*)

Create a discrete type-1 fuzzy set from the given data.

generate_polling_t1_fuzzy_set(*data*)

Create a type-1 fuzzy set with interpolation from the given data.

generate_polling_t2_fuzzy_set(*data*)

Create a type-1 fuzzy set with interpolation from the given data.

generate_iaa_t1_fuzzy_set(*data*)

Create a type-1 interval agreement approach set from interval data.

generate_iaa_t2_fuzzy_set(*data*)

Create a type-2 interval agreement approach set from interval data.

11.2 Variables

Name	Description
__package__	**Value:** `'fuzzycreator'`

12 Module fuzzycreator.global_settings

This module lists settings that are used throughout the toolkit.

12.1 Functions

set_rounding(*decimal_places*)

Set the total decimal values returned in results.

get_x_points()

Get a list of discretised points in the universe of discourse.

get_y_points()

Get a list of discretised alpha-cut points.

get_z_points()

Get a list of discretised zLevels.

get_z_level_heatmap()

Get a list of hex colours for the heat map of zSlices.

get_z_level_greyscale()

Get a list of hex colours for the heat map of zSlices.

12.2 Variables

Name	Description
global_uod	**Value:** `(0, 10)`
global_x_disc	**Value:** `101`
global_alpha_disc	**Value:** `20`
global_zlevel_disc	**Value:** `20`
xlabel	**Value:** `'$x$'`
type_1_ylabel	**Value:** `'$\\mu(x)$'`
type_2_ylabel	**Value:** `'$u(x)$'`
colours	**Value:** `['#006767', '#AC0000', '#E9AF3B', '#34539C', '#E98A3B', '...`
normalise_generated_sets	**Value:** `False`
DECIMAL_ROUNDING	**Value:** `Decimal('0.0001')`

continued on next page

Name	Description
type_2_3d_colour_scheme	**Value:** `0`
GREYSCALE	**Value:** `2`
HEATMAP	**Value:** `1`
UNIQUE	**Value:** `0`
___package___	**Value:** `'fuzzycreator'`

13 Module fuzzycreator.interval_dict

This module is used to create an interval-based dict.

13.1 Variables

Name	Description
___package___	**Value:** `None`

13.2 Class IntervalDict

object
└
fuzzycreator.interval_dict.IntervalDict

This class stores a dict in which the keys are intervals.

13.2.1 Methods

___init___(*self*, *overwrite_with_max*=`True`)

Initiate the interval dict.

When overwrite_with_max = True: If a key has been assigned multiple values the max is returned. When overwrite_with_max = False: If a key has been assigned multiple values the sum is returned.

Overrides: object.___init___

___setitem___(*self*, *_slice*, *_value*)

Assign _value to the continuous _slice.

_slice must be a slice; e.g. [1:3] _value must be numerical

___getitem___(*self*, *_point*)

Return the value of the singleton _point.

singleton_keys(*self*)

Return the list of key values as singletons.

keys(*self*)

Return the list of intervals used as keys.

values(*self*)

Return the list of values stored.

Inherited from object

___delattr___(), ___format___(), ___getattribute___(), ___hash___(), ___new___(), ___reduce___(), ___reduce_ex___(), ___repr___(), ___setattr___(), ___sizeof___(), ___str___(), ___subclasshook___()

13.2.2 Properties

Name	Description
Inherited from object	
___class___	

14 Package fuzzycreator.measures

14.1 Modules

- **distance_gt2**: This module contains distance measures for general type-2 fuzzy set.
  (Section 15, p. 27)
- **distance_it2**: This module contains distance measures for interval type-2 fuzzy set.
  (Section 16, p. 28)
- **distance_t1**: This module contains distance measures for type-1 fuzzy set.
  (Section 17, p. 29)
- **entropy_it2**: This module contains inclusion (subsethood) measures for type-1 sets.
  (Section 18, p. 31)
- **entropy_t1**: This module contains inclusion (subsethood) measures for type-1 sets.
  (Section 19, p. 32)
- **inclusion_t1**: This module contains inclusion (subsethood) measures for type-1 sets.
  (Section 20, p. 33)
- **similarity_gt2**: This module contains similarity measures for general type-2 fuzzy sets.
  (Section 21, p. 34)
- **similarity_it2**: This module contains similarity measures for interval type-2 fuzzy sets.
  (Section 22, p. 35)
- **similarity_t1**: This module contains similarity measures for type-1 fuzzy set.
  (Section 23, p. 36)

14.2 Variables

Name	Description
___package___	**Value:** `None`

15 Module fuzzycreator.measures.distance_gt2

This module contains distance measures for general type-2 fuzzy set.

15.1 Functions

mcculloch(*fs1*, *fs2*)
Calculate the weighted Minkowski (r=1) directional distance.

15.2 Variables

Name	Description
__package__	**Value:** `'fuzzycreator.measures'`

16 Module fuzzycreator.measures.distance_it2

This module contains distance measures for interval type-2 fuzzy set.

16.1 Functions

figueroa_garcia_alpha(*fs1*, *fs2*)

Calculate the absolute difference between alpha-cuts.

figueroa_garcia_centres_hausdorff(*fs1*, *fs2*)

Calculate the hausdorff distance between the centre-of-sets.

figueroa_garcia_centres_minkowski(*fs1*, *fs2*)

Calculate the absolute difference between the centre-of-sets.

mcculloch(*fs1*, *fs2*)

Calculate the weighted Minkowski (r=1) directional distance.

16.2 Variables

Name	Description
__package__	**Value:** `'fuzzycreator.measures'`

17 Module fuzzycreator.measures.distance_t1

This module contains distance measures for type-1 fuzzy set.

17.1 Functions

ralescu1(*fs1*, *fs2*)

Calculate the average Hausdorff distance over all alpha-cuts.

ralescu2(*fs1*, *fs2*)

Calculate the maximum Hausdorff distance over all alpha-cuts.

chaudhuri_rosenfeld(*fs1*, *fs2*)

Calculate the weighted average of Hausdorff distances.

grzegorzewski_non_inf_pq(*fs1*, *fs2*, *p*=2, *q*=0.5)

Grzegorzewski distance where 1 <= p < infty and q is used.

q is used to weight the distance at alpha cuts. (1-q) weight for left distance, (q) weight for right distance.

grzegorzewski_non_inf_p(*fs1*, *fs2*, *p*=2)

Grzegorzewski distance where 1 <= p < infty and q is not used.

grzegorzewski_inf_q(*fs1*, *fs2*, *q*=0.5)

Grzegorzewski distance where p is infinity and q is used.

q is used to weight the distance at alpha cuts. (1-q) weight for left distance, (q) weight for right distance.

grzegorzewski_inf(*fs1*, *fs2*)

Grzegorzewski distance where p is infinity and q is not used.

ban(*fs1*, *fs2*)

Minkowski based distance.

allahviranloo(*fs1*, *fs2*, *c*=0.5, *f*=<__builtin__.function object>)

Distance based on the average width and centre of the fuzzy sets.

yao_wu(*fs1*, *fs2*)

Calculate the average Minkowski (r=1) distance.

mcculloch(*fs1*, *fs2*)

Calculate the weighted Minkowski (r=1) directional distance.

17.2 Variables

Name	Description
__package__	**Value:** 'fuzzycreator.measures'

18 Module fuzzycreator.measures.entropy_it2

This module contains inclusion (subsethood) measures for type-1 sets.

18.1 Functions

szmidt_pacprzyk(*fs*)

Calculate the ratio between the upper & lower membership functions.

zeng_li(*fs*)

Calculate entroyp based on the sum of upper and lower memberships.

18.2 Variables

Name	Description
__package__	**Value:** `'fuzzycreator.measures'`

19 Module fuzzycreator.measures.entropy_t1

This module contains inclusion (subsethood) measures for type-1 sets.

19.1 Functions

kosko(*fs*)

Calculate the degree to which the fuzzy set is fuzzy.

19.2 Variables

Name	Description
___package___	**Value:** `'fuzzycreator.measures'`

20 Module fuzzycreator.measures.inclusion_t1

This module contains inclusion (subsethood) measures for type-1 sets.

20.1 Functions

sanchez(*fs1*, *fs2*)

Calcualte the degree to which fs1 is contained within fs2.

20.2 Variables

Name	Description
___package___	**Value:** `'fuzzycreator.measures'`

21 Module fuzzycreator.measures.similarity_gt2

This module contains similarity measures for general type-2 fuzzy sets.

21.1 Functions

jaccard(*fs1*, *fs2*)

Calculate the weighted average of the jaccard similarity on zslices.

zhao_crisp(*fs1*, *fs2*)

Like jaccard, but the result is the standard average; not weighted.

hao_fuzzy(*fs1*, *fs2*)

Calculate the jaccard similarity given as type-1 fuzzy set.

hao_crisp(*fs1*, *fs2*)

Calculate the centroid of hao_fuzzy(fs1, fs2).

yang_lin(*fs1*, *fs2*)

Calculate the average jaccard similarity for each vertical slice.

mohamed_abdaala(*fs1*, *fs2*)

Based on the the jaccard similarity for each vertical slice.

hung_yang(*fs1*, *fs2*)

Based on the Hausdorff distance between vertical slice pairs.

21.2 Variables

Name	Description
___package___	**Value:** `'fuzzycreator.measures'`

22 Module fuzzycreator.measures.similarity_it2

This module contains similarity measures for interval type-2 fuzzy sets.

22.1 Functions

zeng_li(*fs1*, *fs2*)

Based on the average distance between the membership values.

gorzalczany(*fs1*, *fs2*)

Based on the highest membership where the fuzzy sets overlap.

bustince(*fs1*, *fs2*, *t_norm_min*=`True`)

Based on the inclusion of one fuzzy set within the other.

jaccard(*fs1*, *fs2*)

Ratio between the intersection and union of the fuzzy sets.

zheng(*fs1*, *fs2*)

Similar to jaccard; based on the intersection and union of the sets.

vector(*fs1*, *fs2*)

Vector similarity based on the distance and similarity of shapes.

22.2 Variables

Name	Description
___package___	**Value:** `'fuzzycreator.measures'`

23 Module fuzzycreator.measures.similarity_t1

This module contains similarity measures for type-1 fuzzy set.

23.1 Functions

pappis1(*fs1*, *fs2*)

Based on the maximum distance between membership values.

pappis2(*fs1*, *fs2*)

The ratio between the negation and addition of membership values.

pappis3(*fs1*, *fs2*)

Based on the average difference between membership values.

jaccard(*fs1*, *fs2*)

Ratio between the intersection and union of the fuzzy sets.

dice(*fs1*, *fs2*)

Based on the ratio between the intersection and cardinality.

zwick(*fs1*, *fs2*)

The maximum membership of the intersection of the fuzzy sets.

chen(*fs1*, *fs2*)

Ratio between the product of memberships and the cardinality.

vector(*fs1*, *fs2*)

Vector similarity based on the distance and similarity of shapes.

23.2 Variables

Name	Description
___package___	**Value:** `'fuzzycreator.measures'`

24 Package fuzzycreator.membership_functions

24.1 Modules

- **gaussian**: This module is used to create Gaussian membership functions. *(Section 25, p. 38)*
- **iaa**: This module is for applying the type-1 Interval Agreement Approach. *(Section 26, p. 39)*
- **trapezoidal**: This module is used to create trapezoidal membership functions. *(Section 27, p. 40)*
- **triangular**: This module is used to create triangular membership functions. *(Section 28, p. 41)*

24.2 Variables

Name	Description
__package__	**Value:** `None`

25 Module fuzzycreator.membership_functions.gaussian

This module is used to create Gaussian membership functions.

25.1 Variables

Name	Description
___package___	**Value:** `'fuzzycreator.membership_functions'`

25.2 Class Gaussian

Create a Gaussian distribution.

25.2.1 Methods

___init___(*self*, *mean*, *std_dev*, *height*=`1`)

Set the Gaussian membership function.

height scales the height of the mean.

calculate_membership(*self*, *x*)

Calculate the membership of x. Returns a Decimal value.

calculate_alpha_cut(*self*, *alpha*)

Calculate the alpha-cut of the function.

alpha must be greater than 0 and less than the function height. Returns a two-tuple.

shift_membership_function(*self*, *x*)

Move the membership function along the x-axis by x-amount.

26 Module fuzzycreator.membership_functions.iaa

This module is for applying the type-1 Interval Agreement Approach.

Details of the interval agreement approach are within C. Wagner, S. Miller, J. M. Garibaldi, D. T. Anderson and T. C. Havens, "From Interval-Valued Data to General Type-2 Fuzzy Sets," in IEEE Transactions on Fuzzy Systems, vol. 23, no. 2, pp. 248-269, April 2015. doi: 10.1109/TFUZZ.2014.2310734

26.1 Variables

Name	Description
__package__	**Value:** `'fuzzycreator.membership_functions'`

26.2 Class IntervalAgreementApproach

This class type-1 interval agreement approach membership function.

26.2.1 Methods

__init__(*self*, *normalise*=`False`)

Create a membership function by the interval agreement approach.

add_interval(*self*, *interval*)

Add an interval to the fuzzy set.

calculate_membership(*self*, *x*)

Calculate the membership of x. Returns a Decimal value.

calculate_alpha_cut(*self*, *alpha*)

Calculate the alpha-cut of the function.

alpha must be greater than 0 and less than the function height. Returns a list containing two-tuples (a list of cuts is always given as any alpha-cut may be non-convex)

27 Module fuzzycreator.membership_functions.trapezoidal

This module is used to create trapezoidal membership functions.

27.1 Variables

Name	Description
___package___	**Value:** `'fuzzycreator.membership_functions'`

27.2 Class Trapezoidal

Known Subclasses: fuzzycreator.membership_functions.triangular.Triangular

Create a trapezoidal membership function.

27.2.1 Methods

___init___(*self*, *x_min*, *x_top_left*, *x_top_right*, *x_max*, *height*=`1`)

Set the Trapezoidal membership function.

x_min_base: bottom left coordinate x_top_left: top left coordinate x_top_right: top right coordinate x_max_base: bottom right coordinate height: scale the maximum membership value

calculate_alpha_cut(*self*, *alpha*)

Calculate the alpha-cut of the function.

alpha must be greater than 0 and less than the function height. Returns a two-tuple.

calculate_membership(*self*, *x*)

Calculate the membership of x. Returns a Decimal value.

shift_membership_function(*self*, *x*)

Move the membership function along the x-axis by x-amount.

28 Module fuzzycreator.membership_functions.triangular

This module is used to create triangular membership functions.

28.1 Variables

Name	Description
__package__	**Value:** `'fuzzycreator.membership_functions'`

28.2 Class Triangular

fuzzycreator.membership_functions.trapezoidal.Trapezoidal
└ **fuzzycreator.membership_functions.**

Create a triangular membership function.

28.2.1 Methods

__init__(*self*, *x_min*, *centre*, *x_max*, *height*=1)

Create a triangular membership function.

x_min: bottom left coordinate centre: x coordinate at the peak of the triangle
x_max: bottom right coordinate height: highest membership at the centre.

Overrides:
fuzzycreator.membership_functions.trapezoidal.Trapezoidal.__init__

Inherited from fuzzycreator.membership_functions.trapezoidal.Trapezoidal(Section 27.2)

calculate_alpha_cut(), calculate_membership(), shift_membership_function()

29 Module fuzzycreator.visualisations

This module is used to plot graphs of fuzzy sets.

29.1 Functions

plot_sets(*fuzzy_sets*, *filename*=`None`)

Plot the given list of fuzzy sets.

The fuzzy sets may be of any type. Discretisations and axis labels are set in the global_settings module. If filename is None, the plot is displayed. If a filename is given, the plot is saved to the given location.

29.2 Variables

Name	Description
style	**Value:** `['-', '-']`
___package___	**Value:** `'fuzzycreator'`

30 Module fuzzycreator.visualisations_3d

This module is used to plot 3-dimensional graphs of type-2 fuzzy sets.

30.1 Functions

plot_sets(*fuzzy_sets*, *filename*=`None`)

Display a 3-dimensional plot of the given list of fuzzy sets.

If filename is None, the plot is displayed. If a filename is given, the plot is saved to the given location.

30.2 Variables

Name	Description
___package___	**Value:** `'fuzzycreator'`

Index