Type2Fuzzy Library Implementation: Mendel, Jerry M., and RI Bob John. "Type-2 fuzzy sets made simple." IEEE Transactions on fuzzy systems 10.2 (2002): 117-127.

Carmel Gafa

April 17, 2019

Abstract

Type-2 Fuzzy Sets made simple is undoubted an excellent introduction to Type-2 fuzzy sets and logic. The paper outlines all the definitions and concepts that are necessary to work with type-2 fuzzy sets in a clear and concise manner. This paper illustrates the implementation of all the examples found in the paper by Mendel and John using the type2fuzzy library.

1 Type-2 fuzzy set definition

The paper illustrates several type-2 fuzzy sets concepts with a simple general type-2 fuzzy set,

(0.9/0 + 0.8/0.2+ 0.7/0.4 + 0.6/0.6 + 0.5/0.8)/1
+(0.5/0 + 0.35/0.2 + 0.35/0.4 + 0.2/0.6 + 0.5/0.8)/2
+(0.35/0.6 + 0.35/0.8)/3
+(0.1/0 + 0.35/0.2 + 0.5/0.4 + 0.1/0.6 + 0.35/0.8)/4
+(0.35/0 + 0.5/0.2 + 0.1/0.4 + 0.2/0.6 + 0.2/0.8)/5

This set will be used in this exercise as in the paper.

A type-2 fuzzy set $\tilde{A}$ can be expressed as

$$\tilde{A} = \int_{x\in X} \int_{u \in J_x} \mu_{\tilde{A}}(x,u)/(x,u) \tag{1}$$

where $J_x \subseteq [0,1]$

The following code snippet illustrates how a general type-2 fuzzy set is defined and used, as explained in Example 1 of the original paper.

```python
from type2fuzzy import GeneralType2FuzzySet

'''
Example 1 : definition of the general type-2 fuzzy set
'''

gt2fs_rep =    ''' (0.9/0 + 0.8/0.2+ 0.7/0.4 + 0.6/0.6 + 0.5/0.8)/1
+(0.5/0 + 0.35/0.2 + 0.35/0.4 + 0.2/0.6 + 0.5/0.8)/2
+(0.35/0.6 + 0.35/0.8)/3
+(0.1/0 + 0.35/0.2 + 0.5/0.4 + 0.1/0.6 + 0.35/0.8)/4
+(0.35/0 + 0.5/0.2 + 0.1/0.4 + 0.2/0.6 + 0.2/0.8)/5'''

# create set
gt2fs = GeneralType2FuzzySet.from_representation(gt2fs_rep)
print(f'\nSet representation: {gt2fs}')
```

Set representation: (0.9000 / 0.0000 + 0.8000 / 0.2000 + 0.7000 / 0.4000 + 0.6000 / 0.6000 + 0.5000 / 0.8000) / 1.0000
+ (0.5000 / 0.0000 + 0.3500 / 0.2000 + 0.3500 / 0.4000 +
0.2000 / 0.6000 + 0.5000 / 0.8000) / 2.0000
+ (0.3500 / 0.6000 + 0.3500 / 0.8000) / 3.0000 + (0.1000 / 0.0000
+ 0.3500 / 0.2000 + 0.5000 / 0.4000 +
0.1000 / 0.6000 + 0.3500 / 0.8000) / 4.0000 +
(0.3500 / 0.0000 + 0.5000 / 0.2000 + 0.1000 / 0.4000 +
0.2000 / 0.6000 + 0.2000 / 0.8000) / 5.0000

2 Vertical Slice

A vertical slice is Type-1 fuzzy set $\mu_{\tilde{A}}(x = x', u)$ for $x \in X$ and $\forall u \in J_{x'} \subseteq [0, 1]$, that is:

$$\mu_{\tilde{A}}(x = x', u) = \int\limits_{u \in J_{x'}} f_{x'}(u)/u \tag{2}$$

where $0 \leq f_{x'}(u) \leq 1$

The following code snippet illustrates two methods by which a vertical slice can be obtained. It illustrates the second part of example 1 that is found in the original paper.

```
# different ways to get vertical slice
print('mu_a_tilde(',1,')= ', gt2fs.vertical_slice(1))
print('mu_a_tilde(',2,')= ', gt2fs[2])
print('mu_a_tilde(',3,')= ', gt2fs.vertical_slice(3))
print('mu_a_tilde(',4,')= ', gt2fs[4])
```

```
mu( 1 )= 0.900/0.000 + 0.800/0.200 + 0.700/0.400 + 0.600/0.600 + 0.500/0.800
mu( 2 )=0.500/0.000 + 0.350/0.200 + 0.350/0.400 + 0.200/0.600 + 0.500/0.800
mu( 3 )=0.350/0.600 + 0.350/0.800
mu( 4 )=0.100/0.000 + 0.350/0.200 + 0.500/0.400 + 0.100/0.600 + 0.350/0.800
```

3 Primary Membership

The **domain** of a secondary membership function is called the **primary membership** of x. Hence in

$$\tilde{A} = \int_{x \in X} \mu_{\tilde{A}}(x)/x = \int_{x \in X} \left[\int_{u \in J_x} f(u)/u \right] /x$$

J_x is the primary membership function, where $J_x \subseteq [0, 1]$ for $\forall x \in X$

The code below illustrates the final part of Example 1 where the primary memberships of the general type-2 fuzzy set are listed:

```
# get the primary memberships of the set
# example 1 (continued)
print('\nPrimary Membership:')
for x_k in gt2fs.primary_domain():
print('J_',x_k, ' : ', gt2fs.primary_membership(x_k))
```

```
Primary Membership:
J1.0 : [0.0, 0.2, 0.4, 0.6, 0.8]
J2.0 : [0.0, 0.2, 0.4, 0.6, 0.8]
J3.0 : [0.6, 0.8]
J4.0 : [0.0, 0.2, 0.4, 0.6, 0.8]
J5.0 : [0.0, 0.2, 0.4, 0.6, 0.8]
```

4 Secondary Grade

The **amplitude** of a secondary membership function is the **secondary grade**. Hence in

$$\tilde{A} = \int_{x \in X} \mu_{\tilde{A}}(x)/x = \int_{x \in X} \left[\int_{u \in J_x} f(u)/u \right] /x$$

where $J_x \subseteq [0, 1]$, $f(u)$ is the secondary grade.

The following code illustrates the retrieval of selected secondary grade values from the general type-2 fuzzy set

```
# get the secondary grade of some values
# example 1 (continued)
print('\nSecondary grade of some points:')
print('mu(1,0.2)=', gt2fs.secondary_grade(1, 0.2), '-- should be 0.8')
print('mu(2,0)=', gt2fs.secondary_grade(2, 0), '-- should be 0.5')
print('mu(3,0.8)=', gt2fs.secondary_grade(3, 0.8), '-- should be 0.35')
print('mu(4,0.4)=', gt2fs.secondary_grade(4, 0.4), '-- should be 0.5')
```

Secondary grade of some points:
mu(1,0.2)= 0.8 – should be 0.8
mu(2,0)= 0.5 – should be 0.5
mu(3,0.8)= 0.35 – should be 0.35
mu(4,0.4)= 0.5 – should be 0.5

5 Footprint of Uncertainty

The 2D support of μ is called **the footprint of uncertainty (FOU)**

$$FOU(\tilde{A}) = \{(x, u) \in X \times [0, 1] | \mu_{\tilde{A}}(x, u) > 0\} \tag{3}$$

FOU represents the uncertainty in the primary memberships of $\tilde{A}$. It is the union of all primary memberships

$$FOU(\tilde{A}) = \bigcup_{x \in X} J_x \tag{4}$$

The FOU can be retrieved using a single line of type2fuzzy library code;

```
# get the footprint of uncertainty for the set
footprint = gt2fs.footprint_of_uncertainty()
print('\nFootprint of uncertainty: ', footprint)
```

Footprint of uncertainty: {
1.0: CrispSet([0.00000, 0.80000]),
2.0: CrispSet([0.00000, 0.80000]),
3.0: CrispSet([0.60000, 0.80000]),
4.0: CrispSet([0.00000, 0.80000]),
5.0: CrispSet([0.00000, 0.80000])}

6 Embedded Type-2 Fuzzy Sets

For discrete universes of discourse X and U, an **embedded type-2 set** $\tilde{A}_e$ has N elements, where $\tilde{A}_e$ has exactly one element from $J_{x_1}, J_{x_2}, \ldots, J_{x_N}$; namely $u_1, u_2, \ldots, u_N$ each with associated grade namely $f_{x_1}(u_1), f_{x_2}(u_2), \ldots, f_{x_N}(u_N)$, such that:

$$\tilde{A}_e = \sum_{i=1}^{N} \left[f_{x_i}(u_i) \right] / x_i \tag{5}$$

where $u_i \in J_{x_i} \subseteq [0,1]$

Set $\tilde{A}_e$ is embedded in $\tilde{A}$ and there are a total of:

$$Num(\tilde{A}_e) = \prod_{i=1}^{N} M_i \tag{6}$$

In Example 2, the authors depict one of the possible 1250 embedded type-2 fuzzy sets that are possible ffrom the general type-2 fuzzy set. The following code snippets illustrates the method to obtain the number of embedded

```
# number of embedded sets
print('\nNumber of embedded type-2 sets: ',
gt2fs.embedded_type2_sets_count())
```

Number of embedded type-2 sets: 1250

```
# Example 2
# list all embedded sets
count = 0
print('\nShowing first 10 embedded sets:')
for embedded_set in gt2fs.embedded_type2_sets():
print(embedded_set)
count = count+1
if count > 10:
break
```

Showing first 10 embedded sets:
[(0.9, 0.0, 1.0), (0.5, 0.0, 2.0), (0.35, 0.6, 3.0), (0.1, 0.0, 4.0), (0.35, 0.0, 5.0)]
[(0.9, 0.0, 1.0), (0.5, 0.0, 2.0), (0.35, 0.6, 3.0), (0.1, 0.0, 4.0), (0.5, 0.2, 5.0)]
[(0.9, 0.0, 1.0), (0.5, 0.0, 2.0), (0.35, 0.6, 3.0), (0.1, 0.0, 4.0), (0.1, 0.4, 5.0)]
[(0.9, 0.0, 1.0), (0.5, 0.0, 2.0), (0.35, 0.6, 3.0), (0.1, 0.0, 4.0), (0.2, 0.6, 5.0)]
[(0.9, 0.0, 1.0), (0.5, 0.0, 2.0), (0.35, 0.6, 3.0), (0.1, 0.0, 4.0), (0.2, 0.8, 5.0)]
[(0.9, 0.0, 1.0), (0.5, 0.0, 2.0), (0.35, 0.6, 3.0), (0.35, 0.2, 4.0), (0.35, 0.0, 5.0)]
[(0.9, 0.0, 1.0), (0.5, 0.0, 2.0), (0.35, 0.6, 3.0), (0.35, 0.2, 4.0), (0.5, 0.2, 5.0)]
[(0.9, 0.0, 1.0), (0.5, 0.0, 2.0), (0.35, 0.6, 3.0), (0.35, 0.2, 4.0), (0.1, 0.4, 5.0)]
[(0.9, 0.0, 1.0), (0.5, 0.0, 2.0), (0.35, 0.6, 3.0), (0.35, 0.2, 4.0), (0.2, 0.6, 5.0)]
[(0.9, 0.0, 1.0), (0.5, 0.0, 2.0), (0.35, 0.6, 3.0), (0.35, 0.2, 4.0), (0.2, 0.8, 5.0)]
[(0.9, 0.0, 1.0), (0.5, 0.0, 2.0), (0.35, 0.6, 3.0), (0.5, 0.4, 4.0), (0.35, 0.0, 5.0)]

Example 3 considers the following general type-2 fuzzy set:

$$(0.5/0.9)/x_1 + (0.2/0.7)/x_1 + (0.9/0.2)/x_1 + (0.6/0.6)/x_2 + (0.1/0.4)/x_2$$

For the sake of this exercise, we assign the values of $x_1 = 1$ and $x_2 = 2$ thus obtaining the following set;

$$(0.5/0.9)/1 + (0.2/0.7)/1 + (0.9/0.2)/1 + (0.6/0.6)/2 + (0.1/0.4)/2$$

The embedded type-2 fuzzy sets are listed using the code below;

```
# Example 3
print('\nEmbedded set listing for general type-2 fuzzy set')
print(str(gt2fs_2))
for embedded_set in gt2fs_2.embedded_type2_sets():
print(embedded_set)
```

```
Embedded set listing for general type-2 fuzzy set
(0.5000 / 0.9000 + 0.2000 / 0.7000 + 0.9000 / 0.2000) / 1.0000
+ (0.6000 / 0.6000 + 0.1000 / 0.4000) / 2.0000
[(0.9, 0.2, 1.0), (0.1, 0.4, 2.0)]
[(0.9, 0.2, 1.0), (0.6, 0.6, 2.0)]
[(0.2, 0.7, 1.0), (0.1, 0.4, 2.0)]
[(0.2, 0.7, 1.0), (0.6, 0.6, 2.0)]
[(0.5, 0.9, 1.0), (0.1, 0.4, 2.0)]
[(0.5, 0.9, 1.0), (0.6, 0.6, 2.0)]
```