Name: Selas Moro   Roll No.: 39   Pid: 182074   BECMPNB

EXPERIMENT 5

Aim: To learn the properties and operations on a Fuzzy set.

Theory:

A fuzzy set, then, is a set containing elements that have varying degrees of membership in the set. If X is a collection of objects denoted by x, then fuzzy set Ã in X is a set of ordered pairs.

$$\tilde{A} = \left\{ \left(x, \mu_{\tilde{A}}(x) \right) \mid x \in X \right\}$$

where X is the Universe of Discourse

$\mu_{\tilde{A}}(x)$ is the membership of element x in set Ã and $0 \le \mu \le 1$

Fuzzy Sets Notation

When the universe is discrete and finite

$$\tilde{A} = \left\{ \frac{\mu_A(x_1)}{x_1}, \frac{\mu_A(x_2)}{x_2}, \ldots \right\} = \sum_i \frac{\mu_A(x_i)}{x_i}$$

When the universe is continuous and infinite

$$\tilde{A} = \int \frac{\mu_A(x)}{x}$$

Example

Create a fuzzy set for integers close to 6 where X = the set of integers

Solution: Ã = {(3, 0. 1), (4, 0. 4), (5, 0. 8), (6, 1), (7, 0. 8), (8, 0. 3), (9, 0. 1)}

Fuzzy Set Operations

Consider two fuzzy sets A and B on the universe X. For a given element x of the universe, the set theoretic operations union, intersection and complement are defined as follows:

Union: The membership function $\mu_{\tilde{A}\cup\tilde{B}}(x)$ of union of two fuzzy sets A and B is defined as:

$$\mu_{\tilde{A}\cup\tilde{B}}(x) = \mu_{\tilde{A}}(x) \vee \mu_{\tilde{B}}(x) = max\left\{\mu_{\tilde{A}}(x)\ ,\mu_{\tilde{B}}(x)\right\}\ , x \text{ E } X$$

Intersection: The membership function $\mu_{\tilde{A}\cap\tilde{B}}(x)$ of intersection of two fuzzy sets A and B is defined as:

$$\mu_{\tilde{A}\cap\tilde{B}}(x) = \mu_{\tilde{A}}(x) \wedge \mu_{\tilde{B}}(x) = min\left\{\mu_{\tilde{A}}(x)\ ,\mu_{\tilde{B}}(x)\right\}\ , x \text{ E } X$$

Complement: The membership function of complement of a normalized fuzzy set A, is defined as:

$$\mu_{\bar{\tilde{A}}}(x) = 1 - \mu_{\tilde{A}}(x) \ , x \text{ E X}$$

De Morgan's laws: De Morgan's laws stated for fuzzy sets, as denoted by these expressions.

$$\overline{\tilde{A} \cup \tilde{B}} = \bar{\tilde{A}} \cap \bar{\tilde{B}}$$
$$\overline{\tilde{A} \cap \tilde{B}} = \bar{\tilde{A}} \cup \bar{\tilde{B}}$$

Properties of Fuzzy Set

1. **Support of Fuzzy Set**
   The support of the fuzzy set A is S(A), which is a crisp set of all x ∈ X such that $\mu\tilde{A}(x) > 0$. The element x in X at which $\mu\tilde{A}(x) = 0.5$ is called crossover point.

2. **Core of Fuzzy Set**
   The core of the fuzzy set A is C(A), which is a crisp set of all x ∈ X such that $\mu\tilde{A}(x)=1$.

3. **α-level set and Strong α-level set**
   α-level set is a crisp set of elements that belong to fuzzy set A atleast to degree α
   $\tilde{A}\ \alpha = \{x \in X \,|\mu\tilde{A}(x) \geq \alpha\}$
   Strong α-level set is defined as
   $\tilde{A}'\ \alpha = \{x \in X \,|\mu\tilde{A}(x) > \alpha\}$

4. **Cardinality**
   The cardinality of fuzzy set A is defined as
   $$\left|\tilde{A}\right| = \sum_{x \in \tilde{A}} \mu_{\tilde{A}}(x)$$
   Relative cardinality is
   $$\left\|\tilde{A}\right\| = \frac{\left|\tilde{A}\right|}{\left|\tilde{X}\right|}$$

5. **Height**
   The height of the fuzzy set A is the largest membership grade of an element in A
   Height (A) = max ($\mu\tilde{A}(x)$)

6. **Normality**
   A fuzzy set X is called normal, if there exist atleast one element x ε X such that $\mu\tilde{A}(x)=1$. A fuzzy set that is not normal is called subnormal.

Experiment:

Implementation:

The following Programming Exercises is perfomed in python in the given link:
https://colab.research.google.com/drive/14dOtyVXJlFX2S0yelDknCPcCX8mftKV_?usp=sharing

Output:

1. **Take two sets as input from user the elements and membership values. If membership value is not in [0,1] generate error message.**

Name: Selas Moro

Roll No: 39

PID: 182074

[1]

```
!pip install -U scikit-fuzzy
import skfuzzy as fuzz
import numpy as np
```

```
Collecting scikit-fuzzy
  Downloading scikit-fuzzy-0.4.2.tar.gz (993 kB)
     |████████████████████████████████| 993 kB 12.4 MB/s
Requirement already satisfied: numpy>=1.6.0 in /usr/local/lib/python3.7/dist-packages (from scikit-fuzzy) (1.19.5)
Requirement already satisfied: scipy>=0.9.0 in /usr/local/lib/python3.7/dist-packages (from scikit-fuzzy) (1.4.1)
Requirement already satisfied: networkx>=1.9.0 in /usr/local/lib/python3.7/dist-packages (from scikit-fuzzy) (2.6.3)
Building wheels for collected packages: scikit-fuzzy
  Building wheel for scikit-fuzzy (setup.py) ... done
  Created wheel for scikit-fuzzy: filename=scikit_fuzzy-0.4.2-py3-none-any.whl size=894089 sha256=6bc08ceaf6119b12216061157820b6d5de237812f28c7a1776c829515e7b080f
  Stored in directory: /root/.cache/pip/wheels/d5/74/fc/38588a3d2e3f34f74588e6daa3aa5b0a322bd6f9420a707131
Successfully built scikit-fuzzy
Installing collected packages: scikit-fuzzy
Successfully installed scikit-fuzzy-0.4.2
```

Take two sets as input from user the elements and membership values. If membership value.

[2]

```
#2 3 4 5 6
#1 0.5 0.6 0.2 0.6
#0.5 0.8 0.4 0.7 0.3
x = np.array(list(map(int,input("Elements of x: ").split())))
mfx = np.array(list(map(float,input("Membership value of x: ").split())))
y = np.array(list(map(int,input("Elements of y: ").split())))
mfy = np.array(list(map(float,input("Membership value of y: ").split())))

for i in mfx:
  if i<0 or i>1:
    print('Invalid membership value of x')
    break
for i in mfy:
  if i<0 or i>1:
    print('Invalid membership value of y')
    break
```

```
Elements of x: 2 3 4 5 6
Membership value of x: 1 0.5 0.6 0.2 0.6
Elements of y:  2 3 4 5 6
Membership value of y: 0.5 0.8 0.4 0.7 0.3
```

2. **Perform Union, Intersection, Complement of both sets, Difference of both sets.**

Intersection

[3]

```
# creating an array with integer type
x = np.array([1, 2, 3])
y = np.array([1, 2, 3])
mfx = np.array([0.1, 0.2, 0.3])
mfy = np.array([0.5, 0.1, 0.9])
z,mfz=fuzz.fuzzymath.fuzzy_and(x,mfx,y,mfy)
```

[4]

```
print(z,mfz)
```

```
[1 2 3] [0.1 0.1 0.3]
```

Union

[5]

```
x = np.array([1, 2, 3])
y = np.array([1, 2, 3])
mfx = np.array([0.1, 0.2, 0.3])
mfy = np.array([0.5, 0.1, 0.9])
z,mfz=fuzz.fuzzymath.fuzzy_or(x,mfx,y,mfy)
```

[6]

```
print(z,mfz)
```

```
[1 2 3] [0.5 0.2 0.9]
```

Compliment

[7]

```
x = np.array([1, 2 ])
y = np.array([1, 2])
mfx = np.array([0.1, 0.2, 0.3])
mfy = np.array([0.5, 0.1, 0.9])
z,mfz=fuzz.fuzzymath.fuzzy_not(x)
```

3. **Find and display Support, Core, Height, Cardinality, Relative Cardinality, Alpha Cuts and Strong Alpha Cuts for both Sets:**

Support

[23]
```
x = np.array([ 2, 3,4,5,6])
y = np.array([2,3,4,5,6])
mfx = np.array([1, 0.5 ,0.6, 0.2, 0.6])
mfy = np.array([0.5, 0.8, 0.4, 0.7, 0.3])
support = [x[i] for i in range(len(mfx)) if mfx[i]>0]
print('SUPPORT OF X:',support)
support = [y[i] for i in range(len(mfy)) if mfy[i]>0]
print('Support of y: ',support)
```

```
SUPPORT OF X: [2, 3, 4, 5, 6]
Support of y:  [2, 3, 4, 5, 6]
```

[24]
```
crossover = [x[i] for i in range(len(mfx)) if mfx[i]==0.5]
print(crossover)
```

```
[3]
```

Core

[25]
```
corex = [x[i] for i in range(len(mfx)) if mfx[i]==1]
print('Core of x: ',corex)
corey = [y[i] for i in range(len(mfy)) if mfy[i]==1]
print('Core of y: ',corey)
```

```
Core of x:  [2]
Core of y:  []
```

Height

[26]
```
height = [x[i] for i in range(len(mfx)) if mfx[i]==max(mfx)]
print('Height of x: ',height)
height = [y[i] for i in range(len(mfy)) if mfy[i]==max(mfy)]
print('Height of y: ',height)
```

```
Height of x:  [2]
Height of y:  [3]
```

Singleton Set

[27]
```
if len(corex)==1:
  print('Fuzzy set is Fuzzy Singleton')
```

```
Fuzzy set is Fuzzy Singleton
```

Relative Cardinality and Cardinality:

Cardinality

[30]

```
print("Cardinality of x : ",end=" ")
print(round(sum(mfx),2))
print("Cardinality of y : ",end=" ")
print(round(sum(mfy),2))
```

```
Cardinality of x :  2.9
Cardinality of y :  2.7
```

Relative Cardinality

[31]

```
print("Relative Cardinality of x : ",end=" ")
print(round(sum(mfx)/len(mfx),2))
print("Relative Cardinality of y : ",end=" ")
print(round(sum(mfy)/len(mfy),2))
```

```
Relative Cardinality of x :  0.58
Relative Cardinality of y :  0.54
```

Alpha Cut and Strong Alpha cut:

Alpha Cut

[28]

```
a = float(input('Enter alpha value: '))
alpha_cut = [x[i] for i in range(len(mfx)) if mfx[i]>=a]
print('Alpha-cut of x: ',alpha_cut)
alpha_cut = [y[i] for i in range(len(mfy)) if mfy[i]>=a]
print('Alpha-cut of y: ',alpha_cut)
```

```
Enter alpha value: 0.6
Alpha-cut of x:  [2, 4, 6]
Alpha-cut of y:  [3, 5]
```

Strong Alpha Cut

```
[29]
a = float(input('Enter alpha value: '))
strong_alpha_cut = [x[i] for i in range(len(mfx)) if mfx[i]>a]
print('Strong-alpha-cut of x: ',strong_alpha_cut)
strong_alpha_cut = [y[i] for i in range(len(mfy)) if mfy[i]>a]
print('Strong-alpha-cut of y: ',strong_alpha_cut)
```

```
Enter alpha value: 0.6
Strong-alpha-cut of x:  [2]
Strong-alpha-cut of y:  [3, 5]
```

Post Experiment:

1. **Prove Demorgan's law.**

DE Morgan's Law

```
[32]
z,mfz=fuzz.fuzzymath.fuzzy_or(x,mfx,y,mfy)
mfz1=fuzz.fuzzymath.fuzzy_not(mfz)
print(z,mfz1)
```

```
[2 3 4 5 6] [0.  0.2 0.4 0.3 0.4]
```

```
[33]
mfx1=fuzz.fuzzymath.fuzzy_not(mfx)
mfy1=fuzz.fuzzymath.fuzzy_not(mfy)
z,mfz=fuzz.fuzzymath.fuzzy_and(x,mfx1,y,mfy1)
print(z,mfz)
```

```
[2 3 4 5 6] [0.  0.2 0.4 0.3 0.4]
```

```
[34]
z,mfz=fuzz.fuzzymath.fuzzy_and(x,mfx,y,mfy)
mfz1=fuzz.fuzzymath.fuzzy_not(mfz)
print(z,mfz1)
```

```
[2 3 4 5 6] [0.5 0.5 0.6 0.8 0.7]
```

```
[35]
mfx1=fuzz.fuzzymath.fuzzy_not(mfx)
mfy1=fuzz.fuzzymath.fuzzy_not(mfy)
z,mfz=fuzz.fuzzymath.fuzzy_or(x,mfx1,y,mfy1)
print(z,mfz)
```

```
[2 3 4 5 6] [0.5 0.5 0.6 0.8 0.7]
```

2. **Find and display whether the given sets are normal or subnormal.**

Normal and Subnormal

[36]

```
if len(corex)>=1:
  print('Fuzzy set x is Normal')
else:
  print('Fuzzy set x is Subnormal')
```

```
Fuzzy set x is Normal
```

[38]

```
if len(corey)>=1:
  print('Fuzzy set y is Normal')
else:
  print('Fuzzy set y is Subnormal')
```

```
Fuzzy set y is Subnormal
```

Conclusion:

In this following Experiment, we have being able to learn the concepts of fuzzy sets and implement them in the above link .