Assignment-A1

Title:

Implementation of Fuzzy Operations, Fuzzy relation by Cartesian product of any two fuzzy sets and Implementation of fuzzy relations (Max-Min Composition).

Problem Statement:

Implement Union, Intersection, Complement and Difference operations on fuzzy sets. Also create fuzzy relation by Cartesian product of any two fuzzy sets and perform max-min composition on any two fuzzy relations.

Objectives:

- • Provide an understanding of the basic mathematical elements of fuzzy sets. • Understand and analyse concepts of fuzzy set.
- • To use fuzzy set operations to implement current computing techniques used in fuzzy computing.

Outcomes: The students will be able to

- • Learn mathematical basis as well as the general principles of various soft computing techniques.
- • To analyse the applications using fuzzy set which uses current techniques, skills, and tools necessary for computing.
- • To identity and solves the engineering problems using the fuzzy set theory and identify the differences and similarities between fuzzy sets and classical sets theories.

Hardware/Software Required:

- JAVA/ C/C++/MATLAB/OCTAVE

Theory:

Fuzzy Logic:

Fuzzy logic is an organized method for dealing with imprecise data. It is a multivalued logic that allows intermediate values to be defined between conventional solutions. In classical set theory, the membership of elements in a set is assessed in binary terms according to a bivalent condition — an element either belongs or does not belong to the set. By contrast, fuzzy set theory permits the gradual assessment of the membership of elements in a set; this is described with the aid of a membership function valued in the real unit interval [0, 1]. Bivalent Set Theory can be somewhat limiting if we wish to describe a 'humanistic' problem mathematically.

Fuzzy Sets:

If "X is collection of objects" (named universe of discourse) "denoted generically by x, then a fuzzy set $\tilde{A}=\{(\mu\bar{A}(x))| x \in X\}$ Where: $\mu\bar{A}(x)= X \rightarrow [0,1]$ is called membership function or degree of membership(also, degree of compatibility or degree of truth) of x in A" If the e interval of real numbers [0,1] is replaced by discrete set {0,1}, then fuzzy set A becomes classic or crisp set.

Example: A = {(x1,0.1),(x2,0.7),(x3,1),(x4,0)}

B = {(x1,0.4),(x2,0.3),(x3,1),(x4,0.2)}

A.**<u>Fuzzy Set Operations</u>**

Having two fuzzy sets $\tilde{A}$ and $\tilde{B}$, the universe of information U and an element *y* of the universe, the following relations express the union, intersection and complement operation on fuzzy sets.

1. **Union/Fuzzy 'OR':**

   Let us consider the following representation to understand how the Union/Fuzzy 'OR' relation works

   $$\mu_{AUB}(y) = \mu_A \vee \mu_B \qquad \forall_y \in U$$

   Here ∨ represents the 'max' operation

   Example: A U B = {(x1,0.4),(x2,0.7),(x3,1),(x4,0.2)}

2. **Intersection/Fuzzy 'AND':**

   Let us consider the following representation to understand how the Intersection/Fuzzy 'AND' relation works

   $$\mu_{A\cap B}(y) = \mu_A \wedge \mu_B \qquad \forall_y \in U$$

   Here ∧ represents the 'min' operation.

   Example: A∩B = {(x1,0.1),(x2,0.3),(x3,1),(x4,0)}

3. **Complement/Fuzzy 'NOT':**

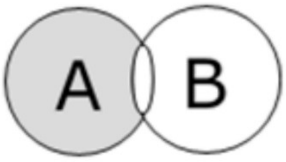
   Let us consider the following representation to understand how the Complement/Fuzzy 'NOT' relation works –

   $$\mu_A = 1- \mu_A(y)$$

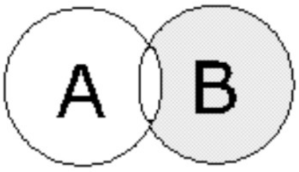
   Example: A^c = {(x1,0.9),(x2,0.3),(x3,0),(x4,1)}

4. **Difference/ Relative Complement:**

   The set difference of sets A and B (denoted by A–B) is the set of elements which are only in A but not in B. Hence, A − B = {x|x ∈ A AND x ∉ B}. Example − If A = {10,11,12,13} and B = {13,14,15}, then (A − B) = {10,11,12} and (B − A) = {14,15}. Here, we can see (A − B) ≠ (B − A)



A - B



B - A

1. **<u>Fuzzy Cartesian product</u>**

The totally ordered set I = [0, 1] is a distributive but not complemented lattice under the operations of infimum ∧ and supremum ∨. On L = I x I we define a partial order ≤, in terms of the partial order on I, as follows: (i) (r1, r2) ≤(s1, s2) iff r1≤s1, r2≤ s2, whenever s1 ≠0 and s2≠0, (ii) (0, 0) = (s1, s2) whenever s1 = 0 or s2 = 0, For every (r1, r2), (s1, s2) ϵ L. The Cartesian product L=IxI is then a distributive but not complemented vector lattice. The operations of minimum and maximum in L are given respectively by (r1, r2) ∧ (s1, s2) = (r1 ∧ s1, r2∧ s2) And (r1, r2) ∨ (s1, s2) = (r1 ∨ s1, r2∨ s2) For every (r1, r2),(s1, s2) ϵ L where the equality holds in the last relation when ri≠0≠si. When we speak of an L-fuzzy subset we mean that the associated membership function takes its values from the lattice L = I x I An L-fuzzy subset of X is thus a function from X to L [4]. The notation {(x, A(x)): x ϵ X} or, simply, {(x, r)}, where r = A(x), will be used to denote a fuzzy subset A of X. Similarly, an L-fuzzy subset of X, a fuzzy subset of X × Y and an L-fuzzy subset of X × Y will be denoted respectively by {(x, (r1, r2))}, {((x, y), r)} and {((x, y), (r1, r2))}. To each fuzzy subset {(x, r1)} of X and fuzzy subset {(y, r2)} of Y there corresponds an L-fuzzy subset {((x, y), (r1, r2))} of X × Y. Throughout this paper the notation (x, r) ϵ A; where A ϵ Ix , will mean that A(x) = r. Definition. The fuzzy Cartesian product of two ordinary sets X and Y, symbolically , is the collection of all L-fuzzy subsets of X × Y. That is,

$$X \ x\ Y = L^{X*Y}$$

An element of X x Y is then a function C:X × Y→ L, or C = {((x, y), (r1, r2)) : (x, y) ϵ X × Y, (r1, r2) = C(x, y) ϵ L}. The fuzzy Cartesian product of a fuzzy subset A = {(x, r)} of X and a fuzzy subset B = ((y, s)} of Y is the L-fuzzy subset A×B of X × Y defined by: A×B = {((x, y), (A(x), B(y))):x ϵ X, y ϵ Y} =- {((x, y), (r, s))}.
It is clear that A×B is an element of for every A ϵ I^X and B ϵ I^Y .
The above definitions can be generalized for any finite number of sets.

2. **<u>Fuzzy relations</u>**

Max-Min Composition of fuzzy Relations Fuzzy relation in different product space can be combined with each other by the operation called ―Composition‖. There are many composition methods in use , e.g. maxproduct method, max-average method and max-min method. But max-min composition method is best known in fuzzy logic applications. Definition: Composition of Fuzzy Relations Consider two fuzzy relation; R (X ×Y) and S (Y ×Z), then a relation T (X × Z), can be expressed asmax-min composition T = R o S μT (x, z) = max-min [μR (x, y), μS (y, z)] = V [μR (x, y) ^ μS (y, z)] If algebraic product is adopted, then max-product compositions adopted: T = R o S μ_T (x, z) = max [μ_R (x, y) . μ_S (y, z)] = V [μ_R (x, y) . μ_S (y, z)] The max-min composition can be interpreted as indicating the strength of the existence of relation between the elements of X and Z. Calculations of (R o S) are almost similar to matrix multiplication. Crisp relation: Crisp relation is defined on the Cartesian product of two sets. Consider,

$$X \times Y = \{(x,y) | x \in X, y \in Y\}$$

The relation on this Cartesian product will be,

$$\mu_R = 1, (x,y) \in R$$
$$0, (x,y) \notin R$$

Example: Let X={1,4,5} and Y={3,6,7} then for relation R=x<y

$$R = \begin{vmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{vmatrix}$$

Fuzzy Relation:

Let be universal sets then,

R = {((x,y), μ_R (x,y)) | (x,y) ∈ X x Y } is called a fuzzy relation in X x Y ⊆ R.

Example: Let X={1,2,3} and Y={1,2} If, then

$$R = \left\{\frac{e^{-(1-1)^2}}{(1,1)}, \frac{e^{-(1-2)^2}}{(1,2)}, \frac{e^{-(2-1)^2}}{(2,1)}, \frac{e^{-(2-2)^2}}{(2,2)}, \frac{e^{-(3-1)^2}}{(3,1)}, \frac{e^{-(3-2)^2}}{(3,2)}\right\} \qquad R = \begin{bmatrix} 1 & 0.37 \\ 0.37 & 1 \\ 0.02 & 0.37 \end{bmatrix}$$

Max-Min Composition:

Let X,Y and Z be universal sets and let R and Q be relations that relate them as,

R = {(x,y) | x ∈ X, y ∈ Y, R ⊂ X x Y }

Q = {(y,z) | y ∈ Y, z ∈ Z, Q ⊂ Y x Z }

Then S will be a relation that relates elements of X with elements of Z as,

S = R ^{0} Q

S = {(x,y)|x ∈ X, z ∈ Z, S ⊂ X x Z }

Max min composition is then defined as,

$\mu_S(x,y) = \max(\min(\mu_R(x,y), \mu_Q(y,z)))$

Example:

$R = \begin{bmatrix} 0.6 & 0.5 & 0.4 \\ 0.2 & 0.1 & 0.2 \end{bmatrix}$ and $Q = \begin{bmatrix} 0.2 & 0.6 \\ 0.1 & 0.3 \\ 0.7 & 0.5 \end{bmatrix}$ $S = \begin{bmatrix} 0.4 & 0.6 \\ 0.2 & 0.2 \end{bmatrix}$

Conclusion:

The concepts of union, intersection and complement are implemented using fuzzy sets which helped to understand the differences and similarities between fuzzy set and classical set theories. It provides the basic mathematical foundations to use the fuzzy set operations. With the use of fuzzy logic principles max min composition of fuzzy set is calculated which describes the relationship between two or more fuzzy sets.

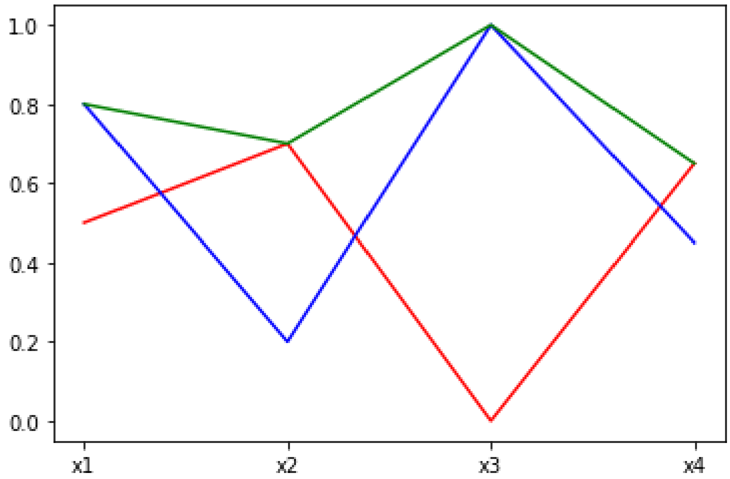
```
import numpy as np
#import skfuzzy as fuzz
from matplotlib import pyplot as plt
```

```
a = {"x1": 0.5, "x2": 0.7, "x3":0.0, "x4":0.65}
b = {"x1": 0.8, "x2": 0.2, "x3":1.0, "x4":0.45}
c = {}
```

```
#UNION
for i in a:
    if a[i]>b[i]:
        c[i]=a[i]
    else:
        c[i]=b[i]

print("Union of Set A and B",c)
x = np.array([0,1,2,3])
y = np.array([(c[e]) for e in c])
my_xticks = np.array([(e) for e in c])
plt.plot(x, np.array([(a[e]) for e in a]), 'r')
plt.plot(x, np.array([(b[e]) for e in b]), 'b')
plt.plot(x, y, 'g')
plt.xticks(x, my_xticks)
plt.show()
```

```
Union of Set A and B {'x1': 0.8, 'x2': 0.7, 'x3': 1.0, 'x4': 0.65}
```

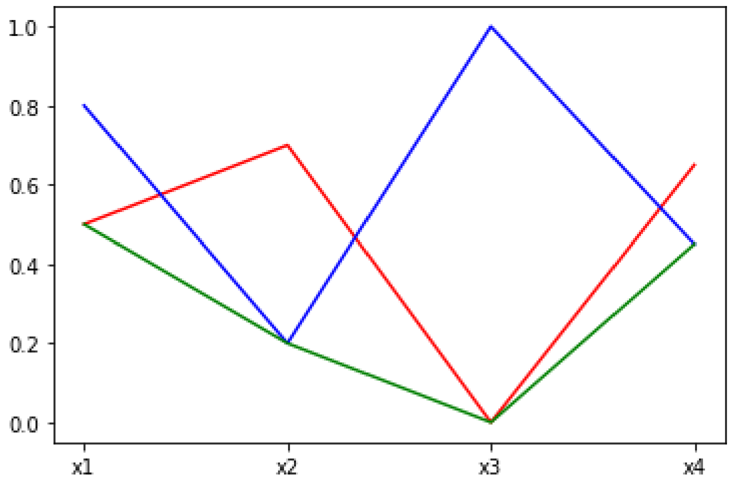



```
#INTERSECTION
d={}
for i in a:
    if a[i]<b[i]:
        d[i]=a[i]
    else:
        d[i]=b[i]

print("Intersection Set A and B",d)
x = np.array([0,1,2,3])
y = np.array([(d[e]) for e in d])
my_xticks = np.array([(e) for e in c])
```

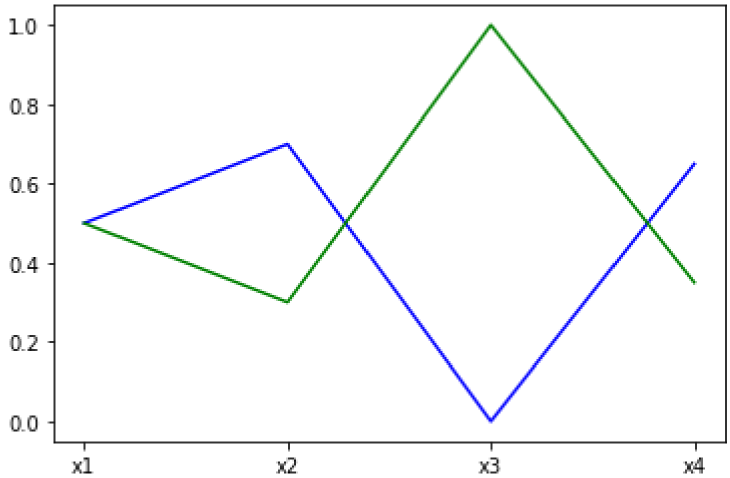
```
my_xticks = np.array([(e) for e in c])
plt.plot(x, np.array([(a[e]) for e in a]), 'r')
plt.plot(x, np.array([(b[e]) for e in b]), 'b')
plt.plot(x, y, 'g')
plt.xticks(x, my_xticks)
plt.show()
```

```
Intersection Set A and B {'x1': 0.5, 'x2': 0.2, 'x3': 0.0, 'x4': 0.45}
```

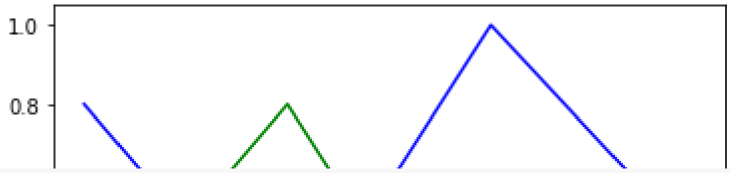



```
#COMPLEMENT
e={}
for i in a:
    e[i]=round(1-a[i],2)
print("Complemet of Set A :" ,e)
x = np.array([0,1,2,3])
y = np.array([(e[i]) for i in e])
my_xticks = np.array([(e) for e in c])
plt.plot(x, np.array([(a[e]) for e in a]), 'b')
plt.plot(x, y, 'g')
plt.xticks(x, my_xticks)
plt.show()
e.clear()
for i in b:
    e[i]=round(1-b[i],2)
print("Complemet of Set B :" ,e)
x = np.array([0,1,2,3])
y = np.array([(e[i]) for i in e])
my_xticks = np.array([(e) for e in c])
plt.plot(x, np.array([(b[e]) for e in b]), 'b')
plt.plot(x, y, 'g')
plt.xticks(x, my_xticks)
plt.show()
```

```
Complemet of Set A : {'x1': 0.5, 'x2': 0.3, 'x3': 1.0, 'x4': 0.35}
```

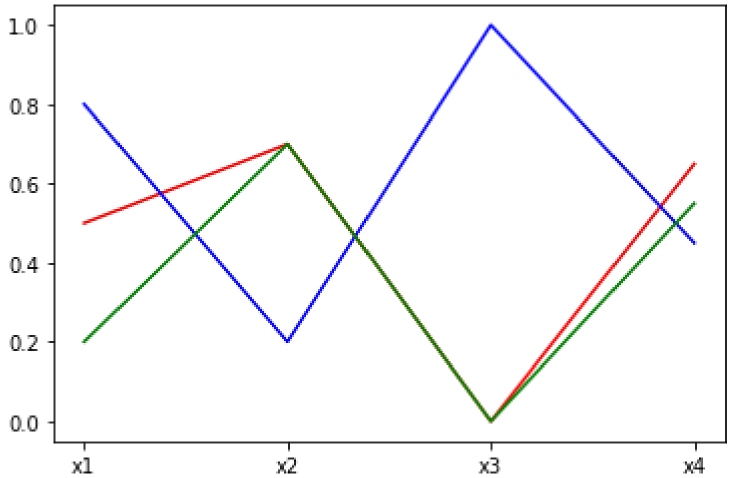



```
Complemet of Set B : {'x1': 0.2, 'x2': 0.8, 'x3': 0.0, 'x4': 0.55}
```




```
#DIFFERNECE
f={}
for i in a:
    f[i]=round((min(a[i],1-b[i])),2)
print(f"Set A: {a} \nSet B: {b}","\nDifference A-B:" ,f)
x = np.array([0,1,2,3])
y = np.array([(f[e]) for e in f])
my_xticks = np.array([(e) for e in c])
plt.plot(x, np.array([(a[e]) for e in a]), 'r')
plt.plot(x, np.array([(b[e]) for e in b]), 'b')
plt.plot(x, y, 'g')
plt.xticks(x, my_xticks)
plt.show()
```

```
Set A: {'x1': 0.5, 'x2': 0.7, 'x3': 0.0, 'x4': 0.65}
Set B: {'x1': 0.8, 'x2': 0.2, 'x3': 1.0, 'x4': 0.45}
Difference A-B: {'x1': 0.2, 'x2': 0.7, 'x3': 0.0, 'x4': 0.55}
```




```
#CARTESIAN PRODUCT
R = [[] for i in range(len(a))]
i = 0
```

```
for x in a:
    for y in b:
        R[i].append(min(a[x], b[y]))
    i += 1
print("R :\n",np.array(R),"\n")
S = [[] for i in range(len(a))]
i = 0
for x in a:
    for y in b:
        S[i].append(min(1-a[x], 1-b[y]))
    i += 1
print("S :\n",np.array(S))
R=np.array(R)
S=np.array(S)
```

```
R :
 [[0.5  0.2  0.5  0.45]
 [0.7  0.2  0.7  0.45]
 [0.   0.   0.   0.  ]
 [0.65 0.2  0.65 0.45]]

S :
 [[0.2  0.5  0.   0.5 ]
 [0.2  0.3  0.   0.3 ]
 [0.2  0.8  0.   0.55]
 [0.2  0.35 0.   0.35]]
```

```
#MAX-MIN OPERATION ON FUZZY RELATION
tmp = np.zeros((R.shape[0], S.shape[1]))
t = list()
for i in range(len(R)):
    for j in range(len(S[0])):
        for k in range(len(S)):
            t.append(round(min(R[i][k], S[k][j]), 2))
        tmp[i][j] = max(t)
        t.clear()
print(tmp)
```

```
[[0.2  0.5  0.   0.5 ]
 [0.2  0.7  0.   0.55]
 [0.   0.   0.   0.  ]
 [0.2  0.65 0.   0.55]]
```