

Práctica 1

Lógica difusa

Operaciones en conjuntos difusos

En esta primera práctica vamos a apoyarnos en Python para visualizar algunas de las operaciones que pueden realizarse con conjuntos difusos.

Sigue las acciones que se proponen a continuación:

1. Abre un “notebook” en Google Colab o bien en Jupyter y ponle como nombre “P1_Operaciones con conjuntos difusos”. Una vez abierta la consola, instala las siguiente librerías de Python¹:

```
import numpy as np
import matplotlib.pyplot as plt
# Scikit-Fuzzy es una colección de algoritmos de lógica difusa
# escritos en Python
import skfuzzy as fuzzb
```

2. Mediante la siguiente línea, creamos el universo del discurso. **¿Qué hace en concreto el método “linspace” del objeto “np” (numpy)?** Búscalo en internet.

```
X = np.linspace(start=0, stop=75, num=75, endpoint=True,
retstep=False)
```

3. Podemos hacer operaciones con conjuntos creando previamente dos funciones de membresía de tipo triangular mediante los métodos que vienen con “scikit”.

```
# Creamos valores para dos conjuntos correspondientes a edades
edades_1 = [0, 25, 50]
edades_2 = [25, 50, 75]

# Creamos sendas funciones de membresía con forma triangular
```

¹Si al importar la librería “[scikit-fuzzy](#)” te da error, seguramente necesitarás instalarla. Puedes con la ayuda del comando “`pip install -U scikit-fuzzy`”.

```
jovenes = fuzzb.membership.trimf(X, abc1)
mediana_edad = fuzzb.membership.trimf(X, abc2)
```

4. Ahora, representamos ambas funciones:

```
plt.figure() # prepara el plano de representación
# Función triangular "Jóvenes"
plt.subplot(3, 2, 1) # ubica la 1ª gráfica en el plano
plt.plot(X, jovenes) # dibuja la 1ª función de membresía
plt.title("Jóvenes")
plt.grid(True)
# Función triangular "Mediana Edad"
plt.subplot(3, 2, 2) # ubica la 2ª gráfica en el plano
plt.plot(X, mediana_edad) # dibuja la 2ª función de membresía
plt.title("Mediana Edad")
plt.grid(True)
plt.show()
```

En las siguientes líneas vamos a representar diferentes operaciones que hemos revisado a través de las diapositivas. Como ejemplo, a continuación tienes la función de unión.

5. Operación de **unión**:

```
# 1. Unión = max( $\mu_A(x)$ ,  $\mu_B(x)$ )
union = fuzzb.fuzzy_or(X, jovenes, X, mediana_edad)[1]

plt.subplot(2, 2, 1)
plt.plot(X, union)
plt.title("Unión")
plt.grid(True)
plt.show()
```

Completa el resto de casos para las diferentes operaciones que se proponen a continuación por tu cuenta, buscando el nombre de los métodos adecuados del objeto "fuzzb" en la [documentación online](#). Representa el resultado por pantalla.

6. Operación de **intersección**:

```
# 2. Intersección =  $\min(\mu_A(x), \mu_B(x))$ 
```

7. Operación de **complemento** (de A):

```
# 3. Complemento (A) =  $(1 - \min(\mu_A(x)))$ 
```

8. Operación de **diferencia** (A/B):

```
# 4. Diferencia (A/B) =  $\min(\mu_A(x), (1 - \mu_B(x)))$ 
```

9. Operación de **producto algebraico**:

```
# 5. Producto algebraico =  $(\mu_A(x) * \mu_B(x))$ 
```