

Міністерство освіти та науки України
Львівський національний університет імені Івана Франка

Факультет електроніки та
Комп'ютерних технологій

Звіт

Про виконання лабораторної роботи №1

“Побудова функцій приналежності нечіткої множини на основі експертної інформації”

Виконав:
Студент групи ФеІ-44
Сапанюк М.І.
Перевірила:
Притула М.

Львів 2022

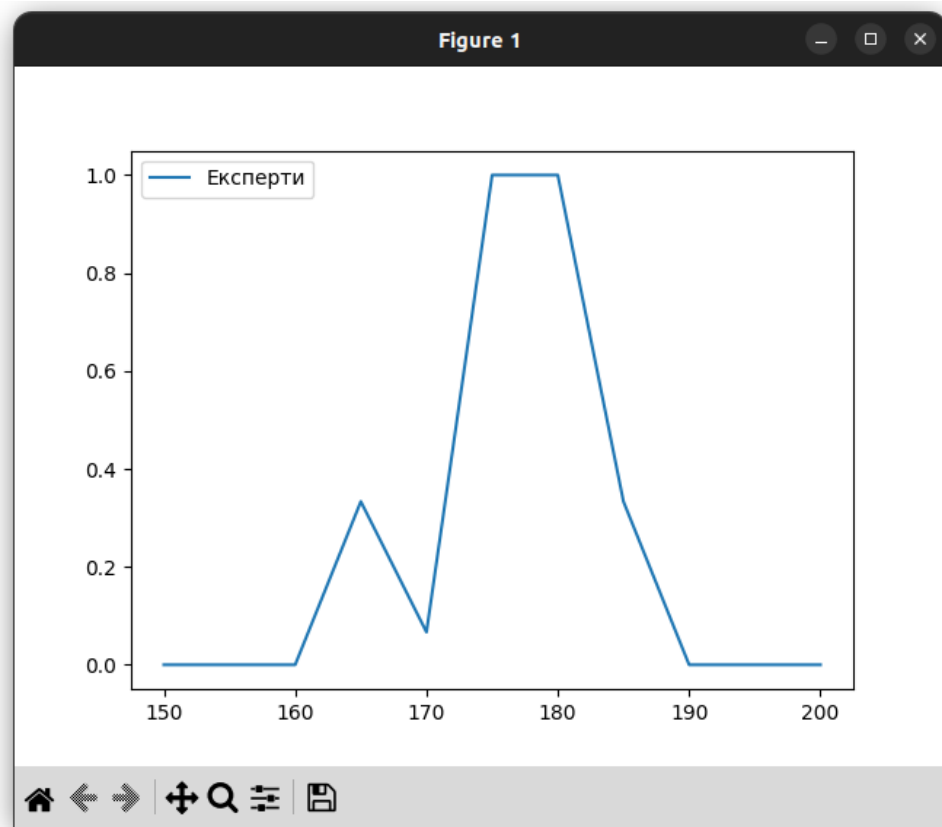
Мета:

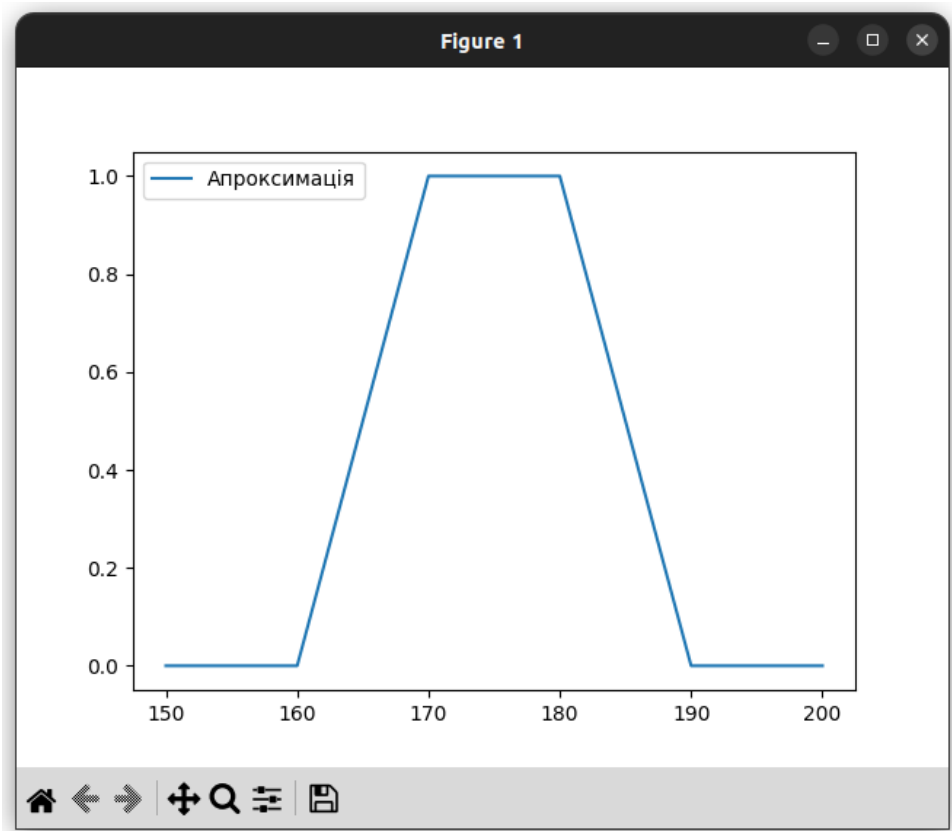
Ознайомитися з прямим методом побудови функцій приналежності нечіткої множини, який ґрунтується на статистичній обробці думок групи експертів.

Хід роботи:

1. На основі опитування 5 або більше експертів задати нечітку множину «чоловік середнього зросту».
2. Створити програму, яка розраховує функцію приналежності на основі статистичної обробки думок групи експертів, буде її графік і апроксимує однією з П-подібних функцій.
3. Визначити носій нечіткої множини, її ядро та межі.

Виконання завдання:





```
Носій
[165, 170, 175, 180, 185]
Ядро
[170, 175, 180]
Межі
[165, 185]
```

Висновок:

Виконавши лабораторну роботу я ознайомився з нечіткими множинами

Додаток:

```
from matplotlib import pyplot as plt
```

```
def aproc(x: int, a: int, b: int, c: int, d: int) -> float:
```

```
    if x < a:
        return 0
    elif a <= x <= b:
        return (x - a) / (b - a)
    elif b <= x <= c:
        return 1
    elif c <= x <= d:
        return (d - x) / (d - c)
    else:
        return 0
```

```
def carrier(array_label: list[int], array_mu: list[float]):
```

```
    temp: list[int] = []
    for i in range(0, len(array_mu)):
        if array_mu[i]:
```

```

        temp.append(array_label[i])
    print("Носій")
    print(temp)

def core(array_label: list[int], array_mu: list[float]):
    temp: list[int] = []
    for i in range(0, len(array_mu)):
        if array_mu[i] == 1:
            temp.append(array_label[i])
    print("Ядро")
    print(temp)

def border(array_label: list[int], array_mu: list[float]):
    temp: list[int] = []
    for i in range(0, len(array_mu)):
        if 0 < array_mu[i] < 1.0:
            temp.append(array_label[i])
    print("Межі")
    print(temp)

a: int = 160
b: int = 170
c: int = 180
d: int = 190
array_label: list[int] = [x for x in range(150, 205, 5)]
array_mu: list[float] = []
array_mu_aprox: list[float] = []
matrix: list[list[float]] = [
    [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
    [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
]
size_i: int = len(matrix)
size_j: int = len(matrix[0])
for j in range(0, size_j):
    sup: float = 0
    mu: float = 0
    for i in range(0, size_i):
        mu += matrix[i][j]
    if sup < matrix[i][j]:
        sup = matrix[i][j]
    if sup:
        mu /= sup * size_i
    array_mu.append(mu)

for x in array_label:
    array_mu_aprox.append(aproc(x, a, b, c, d))
carrier(array_label, array_mu_aprox)
core(array_label, array_mu_aprox)
border(array_label, array_mu_aprox)

plt.plot(array_label, array_mu, label='Експерти')
plt.legend(loc=2)

```

```
plt.show()
plt.plot(array_label, array_mu_aprox, label='Апроксимація')
plt.legend(loc=2)
plt.show()
```