

ЛАБОРАТОРНА РОБОТА №2

Виконавець: студент групи ПК-21м-1
Панасенко Єгор
Сергійович

Постановка задачі

Тема: «Налаштування моделі нечіткого виводу»

Мета роботи: Визначення функцій приналежності на основі навчальної вибірки.

Порядок виконання роботи:

1. Провести процедуру фазифікації для змінних з бази правил (л.р. № 1).
2. Обрати апроксимацію функцій приналежності.
3. Провести генерацію навчальної вибірки на основі бази правил і шкал лінійно-логістичних змінних.
4. Сформулювати функціонал середньоквадратичного відхилення значень, обчислених за допомогою моделі (л.р. № 1), та значень, отриманих з навчальної вибірки.
5. Провести мінімізацію функціоналу одним з градієнтних методів (на вибір).

Хід роботи

Розглянемо модель роботи холодильника у залежності від температури навколишнього середовища, та терміну придатності продукту.

Нехай x_1 — температура навколишнього середовища у градусах Цельсія, x_2 - термін придатності продукту у днях, x_3 — потужність роботи холодильника в діапазоні $[0, 10]$.

Значення нечітких змінних:

- ϕ_1 — низька (Н) або короткий (К)

- ϕ_2 – середня (С)
- ϕ_3 – висока (В) або довгий (Д)

Правила:

- Якщо $x_1 = \text{Н}$ та $x_2 = \text{К}$, то $x_3 = \text{С}$
- Якщо $x_1 = \text{Н}$ та $x_2 = \text{С}$, то $x_3 = \text{Н}$
- Якщо $x_1 = \text{Н}$ та $x_2 = \text{Д}$, то $x_3 = \text{Н}$
- Якщо $x_1 = \text{С}$ та $x_2 = \text{К}$, то $x_3 = \text{В}$
- Якщо $x_1 = \text{С}$ та $x_2 = \text{С}$, то $x_3 = \text{С}$
- Якщо $x_1 = \text{С}$ та $x_2 = \text{Д}$, то $x_3 = \text{Н}$
- Якщо $x_1 = \text{В}$ та $x_2 = \text{К}$, то $x_3 = \text{В}$
- Якщо $x_1 = \text{В}$ та $x_2 = \text{С}$, то $x_3 = \text{В}$
- Якщо $x_1 = \text{В}$ та $x_2 = \text{Д}$, то $x_3 = \text{С}$

Нечітка модель:

- $\mu_{\text{Н}}(x_3) = \max \begin{pmatrix} \min(\mu_{\text{Н}}(x_1), \mu_{\text{С}}(x_2)) \\ \min(\mu_{\text{Н}}(x_1), \mu_{\text{Д}}(x_2)) \\ \min(\mu_{\text{С}}(x_1), \mu_{\text{Д}}(x_2)) \end{pmatrix}$
- $\mu_{\text{С}}(x_3) = \max \begin{pmatrix} \min(\mu_{\text{Н}}(x_1), \mu_{\text{К}}(x_2)) \\ \min(\mu_{\text{С}}(x_1), \mu_{\text{С}}(x_2)) \\ \min(\mu_{\text{В}}(x_1), \mu_{\text{Д}}(x_2)) \end{pmatrix}$
- $\mu_{\text{В}}(x_3) = \max \begin{pmatrix} \min(\mu_{\text{С}}(x_1), \mu_{\text{К}}(x_2)) \\ \min(\mu_{\text{В}}(x_1), \mu_{\text{К}}(x_2)) \\ \min(\mu_{\text{В}}(x_1), \mu_{\text{С}}(x_2)) \end{pmatrix}$

Шкала для змінної виводу

низька	середня	висока
\underline{x}_3	x_3^1	x_3^2
\overline{x}_3		

$$x_3 = \{\underline{x}_3, x_3^1, x_3^2, \overline{x}_3\}$$

Дефазифікація

$$x_3 = \frac{\underline{x}_3 \mu_{\text{Н}}(x_3) + x_3^1 \mu_{\text{С}}(x_3) + x_3^2 \mu_{\text{В}}(x_3) + \overline{x}_3 \mu_{\text{В}}(x_3)}{\mu_{\text{Н}}(x_3) + \mu_{\text{С}}(x_3) + \mu_{\text{В}}(x_3)}$$

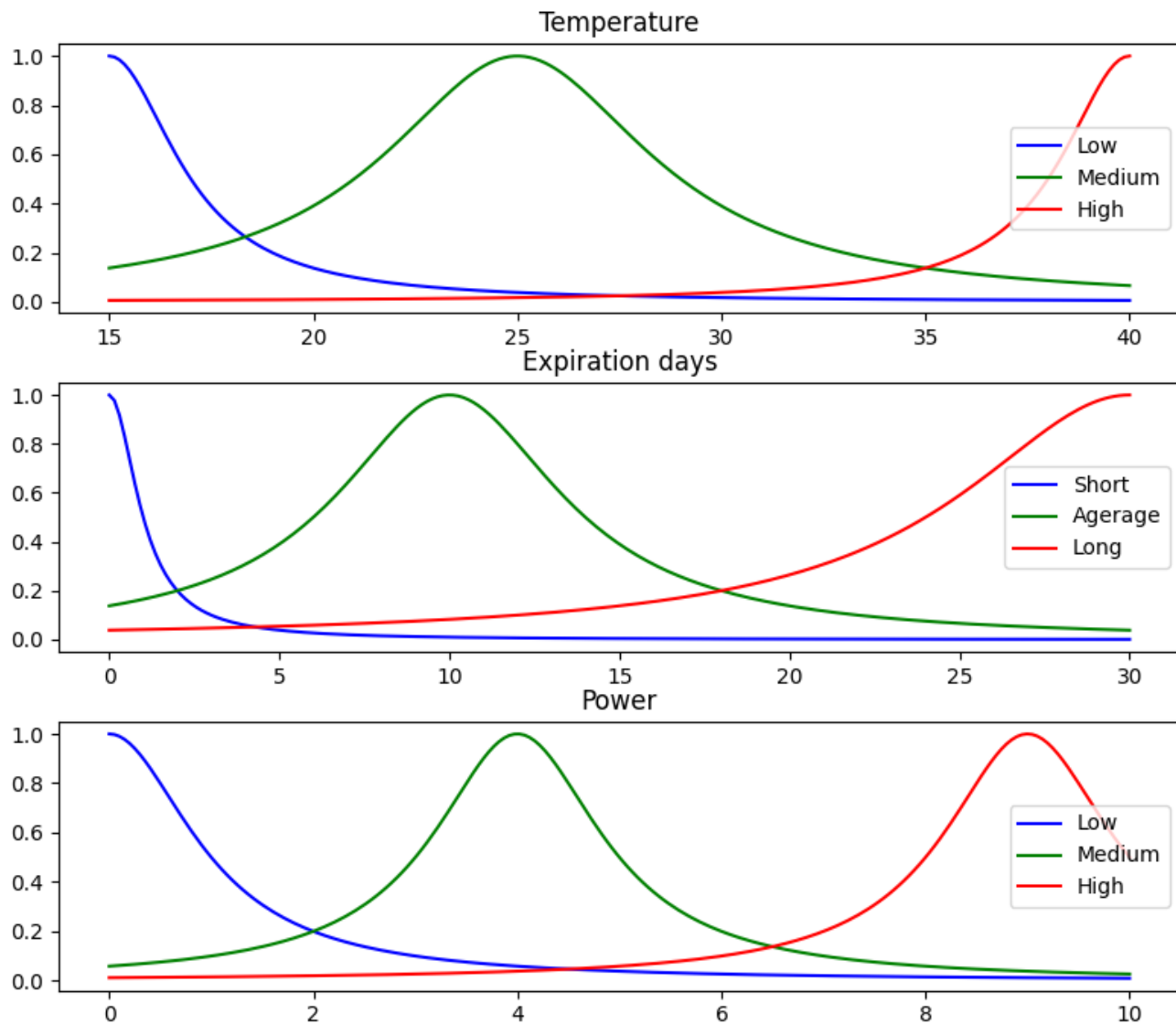
Навчальна вибірка $(y_p^*, x_{ip}^*, p=1, N, i=1, J)$:

x_{1p}^*	x_{2p}^*	y_p^*
15	0	4
15	5	2
15	10	0
15	20	0
15	40	0
20	0	4
20	5	2
20	10	0
20	20	0
20	40	0
25	0	9
25	5	6
25	10	4
25	20	2
25	40	0
32	0	9
32	5	9
32	10	6
32	20	4
32	40	2
40	0	9
40	5	9
40	10	9
40	20	6
40	40	4

Таблиця значень перед налаштуванням:

x_{1p}^*	x_{1p}	$\mu_{Hp}^0(x_{1p}^*)$	$\mu_{Cp}^0(x_{1p}^*)$	$\mu_{Bp}^0(x_{1p}^*)$	$\mu_{Kp}^0(x_{2p}^*)$	$\mu_{Cp}^0(x_{2p}^*)$	$\mu_{Dp}^0(x_{1p}^*)$	\widetilde{y}_p
15	0	1	0.14	0.01	1	0.14	0.04	4.11
15	5	1	0.14	0.01	0.04	0.39	0.05	1.58
15	10	1	0.14	0.01	0.01	1	0.08	0.56
15	20	1	0.14	0.01	0	0.14	0.26	1.49
15	40	1	0.14	0.01	0	0.02	0.26	0.44
20	0	0.14	0.39	0.01	1	0.14	0.04	6.1
20	5	0.14	0.39	0.01	0.04	0.39	0.05	3.37
20	10	0.14	0.39	0.01	0.01	1	0.08	3.07
20	20	0.14	0.39	0.01	0	0.14	0.26	1.55
20	40	0.14	0.39	0.01	0	0.02	0.26	0.54
25	0	0.04	1	0.02	1	0.14	0.04	8.12
25	5	0.04	1	0.02	0.04	0.39	0.05	3.95
25	10	0.04	1	0.02	0.01	1	0.08	3.78
25	20	0.04	1	0.02	0	0.14	0.26	1.69
25	40	0.04	1	0.02	0	0.02	0.26	0.76
32	0	0.01	0.25	0.06	1	0.14	0.04	6.55
32	5	0.01	0.25	0.06	0.04	0.39	0.05	4.21
32	10	0.01	0.25	0.06	0.01	1	0.08	3.91
32	20	0.01	0.25	0.06	0	0.14	0.26	2.44
32	40	0.01	0.25	0.06	0	0.02	0.26	1.22
40	0	0.01	0.07	1	1	0.14	0.04	8.39
40	5	0.01	0.07	1	0.04	0.39	0.05	7.39
40	10	0.01	0.07	1	0.01	1	0.08	8.12
40	20	0.01	0.07	1	0	0.14	0.26	4.9
40	40	0.01	0.07	1	0	0.02	0.26	3.49

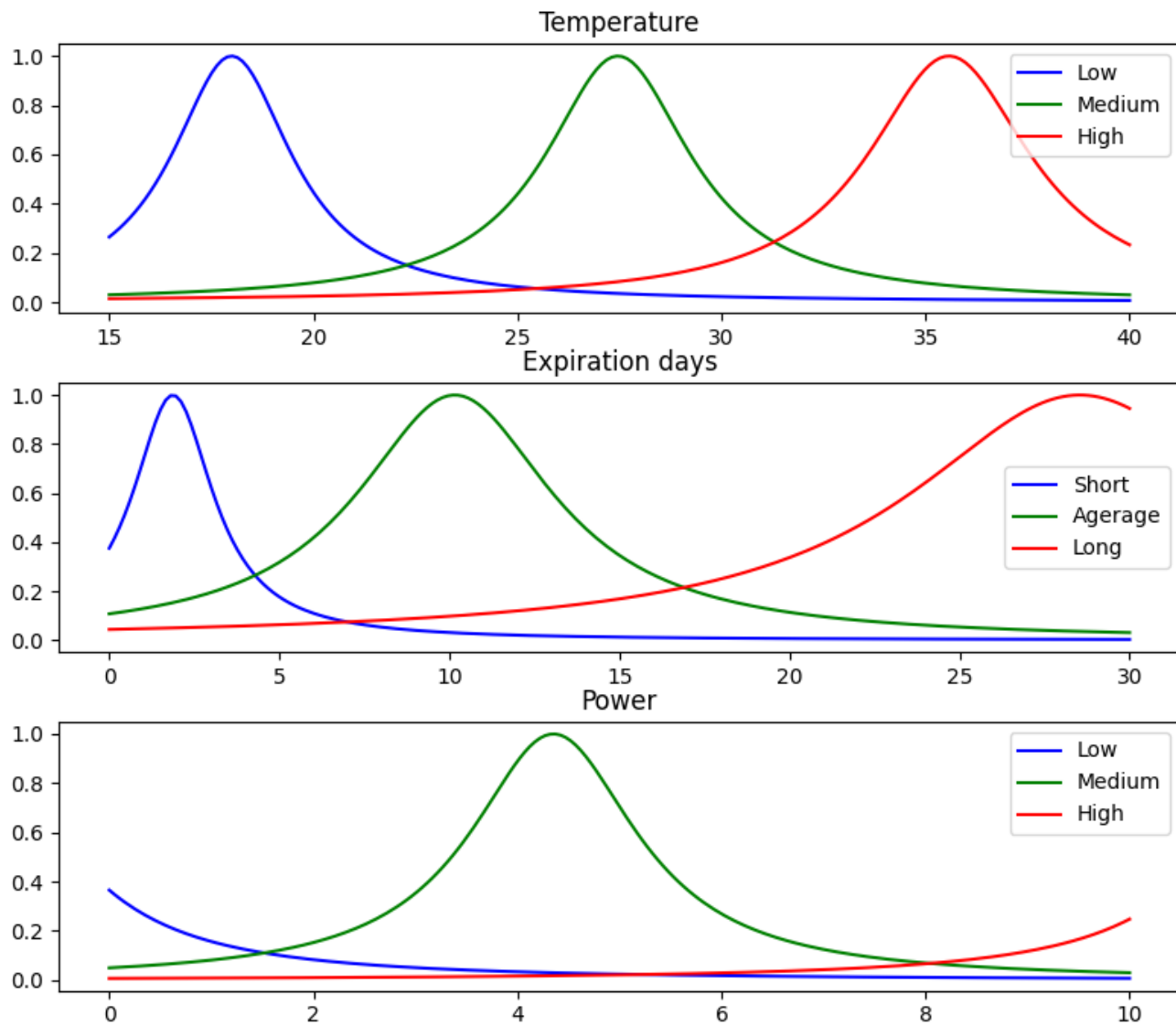
Функції належності термів перед налаштуванням:



Таблиця значень після налаштування:

x_{1p}^*	x_{1p}^*	$\mu_{Hp}^0(x_{1p}^*)$	$\mu_{Cp}^0(x_{1p}^*)$	$\mu_{Bp}^0(x_{1p}^*)$	$\mu_{Kp}^0(x_{2p}^*)$	$\mu_{Cp}^0(x_{2p}^*)$	$\mu_{Dp}^0(x_{1p}^*)$	\widetilde{y}_p
15	0	0.26	0.03	0.01	0.37	0.11	0.04	3.39
15	5	0.26	0.03	0.01	0.18	0.32	0.06	1.63
15	10	0.26	0.03	0.01	0.03	1	0.1	0.41
15	20	0.26	0.03	0.01	0.01	0.11	0.34	-0.19
15	40	0.26	0.03	0.01	0	0.01	0.22	-0.27
20	0	0.45	0.08	0.02	0.37	0.11	0.04	4.3
20	5	0.45	0.08	0.02	0.18	0.32	0.06	2.22
20	10	0.45	0.08	0.02	0.03	1	0.1	0.2
20	20	0.45	0.08	0.02	0.01	0.11	0.34	0.41
20	40	0.45	0.08	0.02	0	0.01	0.22	-0.09
25	0	0.06	0.44	0.05	0.37	0.11	0.04	8.8
25	5	0.06	0.44	0.05	0.18	0.32	0.06	6.06
25	10	0.06	0.44	0.05	0.03	1	0.1	4.05
25	20	0.06	0.44	0.05	0.01	0.11	0.34	1.28
25	40	0.06	0.44	0.05	0	0.01	0.22	0.32
32	0	0.02	0.19	0.32	0.37	0.11	0.04	8.84
32	5	0.02	0.19	0.32	0.18	0.32	0.06	7.86
32	10	0.02	0.19	0.32	0.03	1	0.1	7.34
32	20	0.02	0.19	0.32	0.01	0.11	0.34	3.99
32	40	0.02	0.19	0.32	0	0.01	0.22	2.07
40	0	0.01	0.03	0.23	0.37	0.11	0.04	9.44
40	5	0.01	0.03	0.23	0.18	0.32	0.06	9.14
40	10	0.01	0.03	0.23	0.03	1	0.1	8.68
40	20	0.01	0.03	0.23	0.01	0.11	0.34	6.14
40	40	0.01	0.03	0.23	0	0.01	0.22	4.11

Функції належності термів після налаштування:



Код алгоритму:

```
#!/usr/bin/env python3

import matplotlib.pyplot as plt
import numpy as np
from scipy.optimize import minimize

# Inspiration:
https://pythonhosted.org/scikit-fuzzy/auto\_examples/plot\_tipping\_problem.html

def mf(x, b, c):
    return 1/(1+((x-b)/c)**2)

data = (
    (15, 0, 4),
    (15, 5, 2),
    (15, 10, 0),
    (15, 20, 0),
    (15, 40, 0),

    (20, 0, 4),
    (20, 5, 2),
    (20, 10, 0),
    (20, 20, 0),
    (20, 40, 0),

    (25, 0, 9),
    (25, 5, 6),
    (25, 10, 4),
    (25, 20, 2),
    (25, 40, 0),

    (32, 0, 9),
    (32, 5, 9),
    (32, 10, 6),
    (32, 20, 4),
    (32, 40, 2),

    (40, 0, 9),
    (40, 5, 9),
    (40, 10, 9),
    (40, 20, 6),
    (40, 40, 4),
)

x_temp = np.linspace(15, 40, 200)
x_expr = np.linspace(0, 30, 200)
x_powr = np.linspace(0, 10, 200)

consts = np.array([15, 2, 25, 4, 40, 2, 0, 1, 10, 4, 30, 6, 0, 1, 4, 1, 9, 1],
dtype="float32")

temp_lo_f = lambda x, c: mf(x, c[0], c[1])
temp_md_f = lambda x, c: mf(x, c[2], c[3])
temp_hi_f = lambda x, c: mf(x, c[4], c[5])
expr_sh_f = lambda x, c: mf(x, c[6], c[7])
```



```

expr_md_f = lambda x, c: mf(x, c[8], c[9])
expr_ln_f = lambda x, c: mf(x, c[10], c[11])
powr_lo_f = lambda x, c: mf(x, c[12], c[13])
powr_md_f = lambda x, c: mf(x, c[14], c[15])
powr_hi_f = lambda x, c: mf(x, c[16], c[17])

def mu_lo(x, c):
    return max([
        min(temp_lo_f(x[0], c), expr_md_f(x[1], c)),
        min(temp_lo_f(x[0], c), expr_ln_f(x[1], c)),
        min(temp_md_f(x[0], c), expr_ln_f(x[1], c)),
    ])

def mu_md(x, c):
    return max([
        min(temp_lo_f(x[0], c), expr_sh_f(x[1], c)),
        min(temp_md_f(x[0], c), expr_md_f(x[1], c)),
        min(temp_hi_f(x[0], c), expr_ln_f(x[1], c)),
    ])

def mu_hi(x, c):
    return max([
        min(temp_md_f(x[0], c), expr_sh_f(x[1], c)),
        min(temp_hi_f(x[0], c), expr_sh_f(x[1], c)),
        min(temp_hi_f(x[0], c), expr_md_f(x[1], c)),
    ])

def defuzz(x, c):
    # ~ print(mu_lo(x), mu_md(x), mu_hi(x))
    return (c[12]*mu_lo(x, c)+c[14]*mu_md(x, c)+c[16]*mu_hi(x, c))/(mu_lo(x, c)
+mu_md(x, c)+mu_hi(x, c))

def solve(x, c):
    print("%2.f and %2.f -> %0.3f" % (x[0], x[1], defuzz(x, c)))

def J(c):
    return sum((defuzz([i[0], i[1]], c)-i[2])**2 for i in data)

res = minimize(J, consts, options={'disp': True})
fin_consts = res.x
print(fin_consts)

def show_result(c):
    print()
    for x1, x2, y in data:
        to = temp_lo_f(x1, c)
        tm = temp_md_f(x1, c)
        th = temp_hi_f(x1, c)
        eo = expr_sh_f(x2, c)
        em = expr_md_f(x2, c)
        eh = expr_ln_f(x2, c)
        s = defuzz((x1, x2), c)
        print("%2i,%2i,%2.2f,%2.2f,%2.2f,%2.2f,%2.2f,%2.2f" % (x1, x2, to,
tm, th, eo, em, eh, s))

fig, (ax0, ax1, ax2) = plt.subplots(nrows=3, figsize=(8, 9))

temp_lo = temp_lo_f(x_temp, c)

```

```

temp_md = temp_md_f(x_temp, c)
temp_hi = temp_hi_f(x_temp, c)
expr_sh = expr_sh_f(x_expr, c)
expr_md = expr_md_f(x_expr, c)
expr_ln = expr_ln_f(x_expr, c)
powr_lo = powr_lo_f(x_powr, c)
powr_md = powr_md_f(x_powr, c)
powr_hi = powr_hi_f(x_powr, c)

ax0.plot(x_temp, temp_lo, 'b', linewidth=1.5, label='Low')
ax0.plot(x_temp, temp_md, 'g', linewidth=1.5, label='Medium')
ax0.plot(x_temp, temp_hi, 'r', linewidth=1.5, label='High')
ax0.set_title('Temperature')
ax0.legend()

ax1.plot(x_expr, expr_sh, 'b', linewidth=1.5, label='Short')
ax1.plot(x_expr, expr_md, 'g', linewidth=1.5, label='Agerage')
ax1.plot(x_expr, expr_ln, 'r', linewidth=1.5, label='Long')
ax1.set_title('Expiration days')
ax1.legend()

ax2.plot(x_powr, powr_lo, 'b', linewidth=1.5, label='Low')
ax2.plot(x_powr, powr_md, 'g', linewidth=1.5, label='Medium')
ax2.plot(x_powr, powr_hi, 'r', linewidth=1.5, label='High')
ax2.set_title('Power')
ax2.legend()
plt.tight_layout()

```

```

show_result(consts)
show_result(fin_consts)
plt.show()

```