Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Лабораторна робота №7

Відшукання моделі оптимальної складності методами самоорганізації моделей

Виконала

студентка групи ІТ-91: Луцай К. А.

Перевірив: Іваніщев Б. В.

Київ 2022

Завдання: Для визначення параметрів моделі за даними навчальної послідовності потрібно використати програму, яку було складено на лабораторній роботі «Ідентифікація об‘єкта за даними спостережень».

Варіант 15 (3):

3) Використати функції виду $y = b_0 + b_1\frac{1}{x} + \ldots + b_n\frac{1}{x^n}$ та критерій мінімуму зсуву для пошуку моделі оптимальної складності самоорганізації моделей за такими даними спостережень:

x	1	1,5	2	2,5	3	3,5	4	4,5
y	14	18,222	18	17,216	16,444	15,778	15,219	14,749
x	5	5,5	6	6,5	7	7,5	8	8,5
y	14,352	14,014	13,722	13,469	13,248	13,052	12,879	12,724

Лістинг програми Python:

```
import numpy as np
import matplotlib.pyplot as plt
import itertools

n = 16

X0 = np.zeros(n)
for i in range(n):
    X0[i] = 1 + i * 0.5

Y0 = np.array([14, 18.222, 18, 17.216, 16.444, 15.778, 15.219, 14.749,
               14.352, 14.014, 13.722, 13.469, 13.248, 13.052, 12.879,
12.724])


def func(X, B):
    Y = np.zeros_like(X)
    for n in range(len(X)):
        Y[n] = B[0]
        for i in range(1, len(B)):
            Y[n] += B[i] / np.power(X[n], i)
    return Y


def solve_sle(X, Y, m):
    A = np.zeros((m, m))
    A[:, 0] = 1
```

```
    for k in range(1, m):
        A[:, k] = 1 / np.power(X, k)
    B = np.linalg.solve(A, Y)
    return B


def reg_crit(Y, Y_reg):
    return np.sum(np.square(np.subtract(Y, Y_reg))) /
np.sum(np.square(Y))


def ms_crit(Yt, Ya, Yb):
    yb = np.zeros_like(Ya)
    yb[:len(Yb)] = Yb
    return 3 * np.sum(np.square(np.subtract(Ya, yb))) /
np.sum(np.square(Yt))


def data_split(X, Y, test_step):
    X_test, y_test = X[::test_step], Y[::test_step]
    X_train, y_train = np.delete(X, slice(None, None, test_step)),
np.delete(Y, slice(None, None, test_step))
    print(f"TRAIN SIZE {len(X_train)}\n\tX train {X_train}\n\tY train
{y_train}")
    print(f"TEST SIZE {len(X_test)}\n\tX test {X_test}\n\tY test
{y_test}")
    return X_train, X_test, y_train, y_test


def print_regression(coefs, c):
    formula = "Y = "
    for i in range(c):
        if coefs[i] != 0:
            formula += f"{round(coefs[i], 2)}/X^{i} + "

    formula = formula[:-2]
    print(formula)


def get_model(X, Y):
    split_r_crits, split_ms_crits, split_models = {}, {}, {}
    for test_step in range(2, 7):
        X_train, X_test, y_train, y_test = data_split(X, Y, test_step)
        t = len(X_train)

        C = solve_sle(X_train, y_train, t)
        print(f"Unmasked model coefs: {C}")
        print_regression(C, t)

        coef_mask = np.array(list(map(list, itertools.product([0, 1],
repeat=t))))
        coef_mask = coef_mask[(coef_mask == 1).sum(axis=1) > 1]
        coef_mask = np.flip(coef_mask[np.argsort(coef_mask.sum(axis=1))],
0)
```

```
        print(f"Total mask iterations to check: {coef_mask.shape[0]}")

        r_crits, ms_crits, models = {}, {}, {}
        for model in range(coef_mask.shape[0]):
            C_masked = np.where(coef_mask[model], C, 0)
            Y_reg = func(X_test, C_masked)
            R = reg_crit(y_test, Y_reg)

            if len(r_crits.values()) > 1 and min(r_crits.values()) > R:
                print(f"R {model}\tmask {coef_mask[model]} - crit {R}")

            MS = ms_crit(Y, func(X_train, C_masked), Y_reg)
            if len(ms_crits.values()) > 1 and min(ms_crits.values()) >
MS:
                print(f"MS {model}\tmask {coef_mask[model]} - crit {MS}")

            r_crits[model], ms_crits[model], models[model] = R, MS,
C_masked

        best_ind_r, best_ind_ms = min(r_crits, key=r_crits.get),
min(ms_crits, key=ms_crits.get)
        print(f"BEST MODEL\nR\tmask {coef_mask[best_ind_r]} - crit
{r_crits[best_ind_r]}"
              f"\nMS\tmask {coef_mask[best_ind_ms]} - crit
{ms_crits[best_ind_ms]}")
        split_r_crits[test_step], split_ms_crits[test_step] =
r_crits[best_ind_r], ms_crits[best_ind_ms]
        split_models[test_step] = [models[best_ind_r],
models[best_ind_ms]]

    best_model_ind_r, best_model_ind_ms = min(split_r_crits,
key=split_r_crits.get), min(split_ms_crits, key=split_ms_crits.get)
    Mr, Mms = split_models[best_model_ind_r][0],
split_models[best_model_ind_ms][1]
    print(f"BEST\nR test step {best_model_ind_r}\tmodel{Mr} - crit
{split_r_crits[best_model_ind_r]}"
          f"\nMS test step {best_model_ind_ms}\tmodel{Mms} - crit
{split_ms_crits[best_model_ind_ms]}")
    return best_model_ind_ms, Mms


s, m = get_model(X0, Y0)
X_train, X_test, y_train, y_test = data_split(X0, Y0, s)
t = len(X_train)
print("Final (the best) regression model: ")
print_regression(m, t)
y_reg = func(X_test, m)
r = reg_crit(y_test, y_reg)
ms = ms_crit(Y0, func(X_train, m), y_reg)
print(f"Regression crit: {r}")
print(f"Minimal shift crit: {ms}")
```

Результат виконання:

```
TRAIN SIZE 8

      X train [1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5]

      Y train [18.222 17.216 15.778 14.749 14.014 13.469 13.052 12.724]

TEST SIZE 8

      X test [1. 2. 3. 4. 5. 6. 7. 8.]

      Y test [14.    18.    16.444 15.219 14.352 13.722 13.248 12.879]

Unmasked model coefs: [ 9.90593871e+00  2.96914817e+01 -9.90002728e+01  7.33318214e+02

 -3.53311812e+03  9.26086144e+03 -1.23704288e+04  6.48622162e+03]

Y = 9.91/X^0 + 29.69/X^1 + -99.0/X^2 + 733.32/X^3 + -3533.12/X^4 + 9260.86/X^5 + -
12370.43/X^6 + 6486.22/X^7

Total mask iterations to check: 247

R 4    mask [1 1 1 0 1 1 1 1] - crit 35.404438275480246

MS 4   mask [1 1 1 0 1 1 1 1] - crit 1.9742577907176595

R 13   mask [1 1 0 0 1 1 1 1] - crit 12.48551477045216

MS 23  mask [0 1 1 0 1 1 1 1] - crit 1.9742577907176524

MS 60  mask [1 0 0 0 1 1 1 1] - crit 1.972491144508346

R 177  mask [1 1 1 0 0 0 0 0] - crit 3.298419939324114

MS 177 mask [1 1 1 0 0 0 0 0] - crit 1.7448435933089814

R 231  mask [1 1 0 0 0 0 0 0] - crit 0.41043537126313256

MS 231 mask [1 1 0 0 0 0 0 0] - crit 0.0926386372078789

BEST MODEL

R      mask [1 1 0 0 0 0 0 0] - crit 0.41043537126313256

MS     mask [1 1 0 0 0 0 0 0] - crit 0.0926386372078789

TRAIN SIZE 10

      X train [1.5 2.  3.  3.5 4.5 5.  6.  6.5 7.5 8. ]

      Y train [18.222 18.    16.444 15.778 14.749 14.352 13.722 13.469 13.052 12.879]

TEST SIZE 6

      X test [1.  2.5 4.  5.5 7.  8.5]

      Y test [14.    17.216 15.219 14.014 13.248 12.724]

Unmasked model coefs: [ 1.72885503e+01 -2.65643947e+02  5.00664411e+03 -4.92580025e+04
```

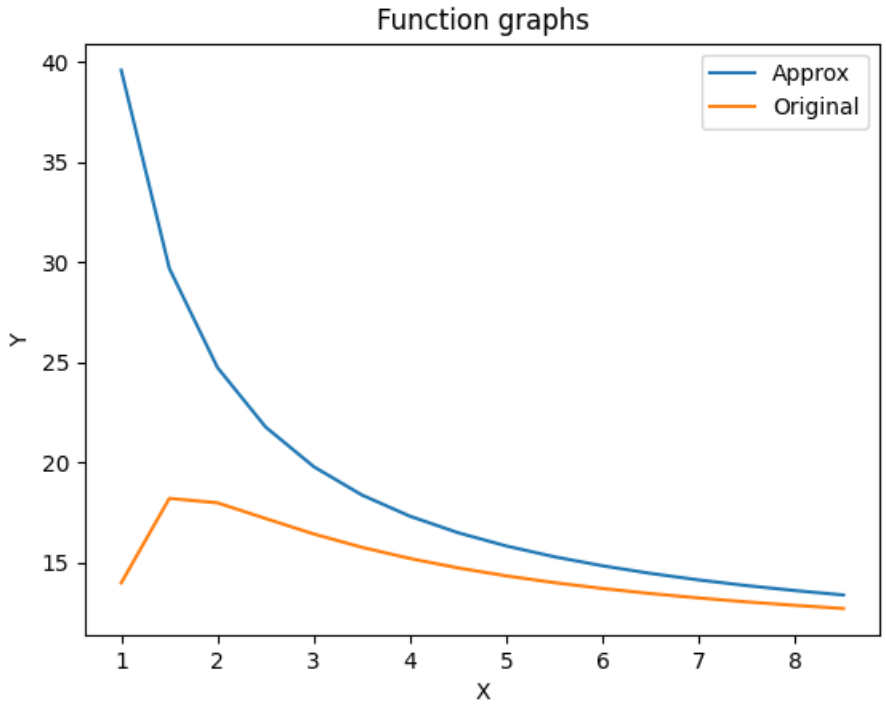
```
  3.01573385e+05 -1.19272366e+06  3.03916929e+06 -4.79775889e+06

  4.24542047e+06 -1.59989864e+06]

Y = 17.29/X^0 + -265.64/X^1 + 5006.64/X^2 + -49258.0/X^3 + 301573.38/X^4 + -
1192723.66/X^5 + 3039169.29/X^6 + -4797758.89/X^7 + 4245420.47/X^8 + -1599898.64/X^9

Total mask iterations to check: 1013

MS 2   mask [0 1 1 1 1 1 1 1 1 1] - crit 2000083.5967760077

R 3    mask [1 0 1 1 1 1 1 1 1 1] - crit 1867519.1650566638

MS 3   mask [1 0 1 1 1 1 1 1 1 1] - crit 1992834.071676598

R 5    mask [1 1 1 0 1 1 1 1 1 1] - crit 8682.703966181563

MS 5   mask [1 1 1 0 1 1 1 1 1 1] - crit 176253.91102771607

R 13   mask [0 1 1 0 1 1 1 1 1 1] - crit 8547.556879805512

MS 13  mask [0 1 1 0 1 1 1 1 1 1] - crit 176235.30792390092

MS 14  mask [1 0 1 0 1 1 1 1 1 1] - crit 174453.79264378292

MS 139 mask [0 0 1 0 1 1 1 1 1 1] - crit 174430.7124041392

MS 949 mask [1 1 1 0 0 0 0 0 0 0] - crit 6386.947805749701

R 979  mask [1 1 0 0 0 0 0 0 0 0] - crit 70.06989680047127

MS 979 mask [1 1 0 0 0 0 0 0 0 0] - crit 10.59960237776371

BEST MODEL

R      mask [1 1 0 0 0 0 0 0 0 0] - crit 70.06989680047127

MS     mask [1 1 0 0 0 0 0 0 0 0] - crit 10.59960237776371

TRAIN SIZE 12

       X train [1.5 2.  2.5 3.5 4.  4.5 5.5 6.  6.5 7.5 8.  8.5]

       Y train [18.222 18.    17.216 15.778 15.219 14.749 14.014 13.722 13.469 13.052
 12.879 12.724]

TEST SIZE 4

       X test [1. 3. 5. 7.]

       Y test [14.    16.444 14.352 13.248]

Unmasked model coefs: [-2.26442177e+02  1.26021534e+04 -2.97462092e+05  4.12310459e+06
 -3.71685687e+07  2.28447764e+08 -9.75244704e+08  2.88660044e+09
 -5.79454715e+09  7.49863466e+09 -5.61922177e+09  1.84380733e+09]

Y = -226.44/X^0 + 12602.15/X^1 + -297462.09/X^2 + 4123104.59/X^3 + -37168568.71/X^4 +
228447763.97/X^5 + -975244703.55/X^6 + 2886600440.0/X^7 + -5794547151.16/X^8 +
7498634658.17/X^9 + -5619221773.9/X^10 + 1843807325.8/X^11
```

```
Total mask iterations to check: 4083
R 9    mask [1 1 1 0 1 1 1 1 1 1 1 1] - crit 1135099026891.2305
MS 9   mask [1 1 1 0 1 1 1 1 1 1 1 1] - crit 875678138715.2601
MS 57  mask [0 1 1 0 1 1 1 1 1 1 1 1] - crit 875678065464.0178
R 63   mask [1 0 1 0 1 1 1 1 1 1 1 1] - crit 1134178827710.6868
MS 63  mask [1 0 1 0 1 1 1 1 1 1 1 1] - crit 875453431444.102
MS 119 mask [0 0 1 0 1 1 1 1 1 1 1 1] - crit 875453351702.2404
MS 2707      mask [1 1 1 1 1 0 0 0 0 0 0 0] - crit 620236496876.622
MS 3366      mask [1 1 0 1 1 0 0 0 0 0 0 0] - crit 612565185114.3281
R 3632 mask [1 1 1 1 0 0 0 0 0 0 0 0] - crit 17391501410.533833
MS 3632      mask [1 1 1 1 0 0 0 0 0 0 0 0] - crit 6452055786.134375
R 3878 mask [1 0 1 1 0 0 0 0 0 0 0 0] - crit 17276237613.298527
MS 3878      mask [1 0 1 1 0 0 0 0 0 0 0 0] - crit 6429942830.205672
R 3944 mask [1 1 1 0 0 0 0 0 0 0 0 0] - crit 96994989.28587496
MS 3944      mask [1 1 1 0 0 0 0 0 0 0 0 0] - crit 24758786.267451905
R 4020 mask [1 1 0 0 0 0 0 0 0 0 0 0] - crit 207701.06975742587
MS 4020      mask [1 1 0 0 0 0 0 0 0 0 0 0] - crit 53088.78784875751
BEST MODEL
R      mask [1 1 0 0 0 0 0 0 0 0 0 0] - crit 207701.06975742587
MS     mask [1 1 0 0 0 0 0 0 0 0 0 0] - crit 53088.78784875751
TRAIN SIZE 12
       X train [1.5 2.  2.5 3.  4.  4.5 5.  5.5 6.5 7.  7.5 8. ]
       Y train [18.222 18.    17.216 16.444 15.219 14.749 14.352 14.014 13.469 13.248
 13.052 12.879]
TEST SIZE 4
       X test [1.  3.5 6.  8.5]
       Y test [14.    15.778 13.722 12.724]
Unmasked model coefs: [ 5.93104705e+02 -2.92593288e+04  6.54431415e+05 -8.58126326e+06
  7.32666536e+07 -4.27150682e+08  1.73288464e+09 -4.88500820e+09
  9.36412067e+09 -1.16077223e+10  8.36191278e+09 -2.64804887e+09]
```

```
Y = 593.1/X^0 + -29259.33/X^1 + 654431.42/X^2 + -8581263.26/X^3 + 73266653.55/X^4 + -427150682.17/X^5 + 1732884643.3/X^6 + -4885008197.7/X^7 + 9364120668.34/X^8 + -11607722304.27/X^9 + 8361912783.34/X^10 + -2648048865.65/X^11

Total mask iterations to check: 4083

R 9    mask [1 1 1 0 1 1 1 1 1 1 1 1] - crit 1551206802121.6104

MS 9   mask [1 1 1 0 1 1 1 1 1 1 1 1] - crit 1195393055904.1233

MS 57  mask [0 1 1 0 1 1 1 1 1 1 1 1] - crit 1195392616814.3125

R 63   mask [1 0 1 0 1 1 1 1 1 1 1 1] - crit 1548628186405.8967

MS 63  mask [1 0 1 0 1 1 1 1 1 1 1 1] - crit 1194797580060.5671

MS 119 mask [0 0 1 0 1 1 1 1 1 1 1 1] - crit 1194797099788.5974

R 3632 mask [1 1 1 1 0 0 0 0 0 0 0 0] - crit 79627327093.95973

MS 3632      mask [1 1 1 1 0 0 0 0 0 0 0 0] - crit 27933678435.74213

R 3878 mask [1 0 1 1 0 0 0 0 0 0 0 0] - crit 79039413140.24109

MS 3878      mask [1 0 1 1 0 0 0 0 0 0 0 0] - crit 27821850935.610146

R 3944 mask [1 1 1 0 0 0 0 0 0 0 0 0] - crit 495358446.87142605

MS 3944      mask [1 1 1 0 0 0 0 0 0 0 0 0] - crit 125104200.60391453

R 4020 mask [1 1 0 0 0 0 0 0 0 0 0 0] - crit 1144175.6521580392

MS 4020      mask [1 1 0 0 0 0 0 0 0 0 0 0] - crit 336356.49884755886

BEST MODEL

R      mask [1 1 0 0 0 0 0 0 0 0 0 0] - crit 1144175.6521580392

MS     mask [1 1 0 0 0 0 0 0 0 0 0 0] - crit 336356.49884755886

TRAIN SIZE 13

       X train [1.5 2.  2.5 3.  3.5 4.5 5.  5.5 6.  6.5 7.5 8.  8.5]

       Y train [18.222 18.    17.216 16.444 15.778 14.749 14.352 14.014 13.722 13.469
 13.052 12.879 12.724]

TEST SIZE 3

       X test [1. 4. 7.]

       Y test [14.    15.219 13.248]

Unmasked model coefs: [-9.01854112e+02  5.26575558e+04 -1.36467229e+06  2.09953416e+07
 -2.13240205e+08  1.50452410e+09 -7.55240455e+09  2.71439336e+10
 -6.92365424e+10  1.22075260e+11 -1.41043081e+11  9.57515157e+10
 -2.88459501e+10]
```

```
Y = -901.85/X^0 + 52657.56/X^1 + -1364672.29/X^2 + 20995341.57/X^3 + -213240205.48/X^4 +
1504524102.95/X^5 + -7552404552.21/X^6 + 27143933591.87/X^7 + -69236542407.72/X^8 +
122075259717.85/X^9 + -141043080523.88/X^10 + 95751515672.94/X^11 + -28845950059.55/X^12

Total mask iterations to check: 8178

R 2    mask [0 1 1 1 1 1 1 1 1 1 1 1] - crit 260400610870328.16

R 6    mask [1 1 1 1 0 1 1 1 1 1 1 1] - crit 55564398300776.125

MS 6   mask [1 1 1 1 0 1 1 1 1 1 1 1] - crit 42863118694341.91

R 42   mask [0 1 1 1 0 1 1 1 1 1 1 1] - crit 55563853597580.77

R 89   mask [1 1 0 1 0 1 1 1 1 1 1 1] - crit 54739328952679.5

MS 89  mask [1 1 0 1 0 1 1 1 1 1 1 1] - crit 42584384532486.01

R 339  mask [0 1 0 1 0 1 1 1 1 1 1 1] - crit 54738788669034.53

R 1002 mask [1 1 1 1 0 0 1 0 1 1 1 1 0] - crit 353963814256.7928

R 1463 mask [1 0 1 1 0 0 1 0 1 1 1 1 0] - crit 351549906493.50757

R 2113 mask [1 1 1 0 0 0 1 0 1 1 1 1 0] - crit 82850777170.81818

R 3160 mask [0 1 1 0 0 0 1 0 1 1 1 1 0] - crit 82823030745.89697

R 3657 mask [1 1 0 0 0 0 1 0 1 1 1 1 0] - crit 55493481918.84607

R 4099 mask [0 1 0 0 0 0 1 0 1 1 1 1 0] - crit 55470155035.04011

MS 6618      mask [1 1 1 1 1 0 0 0 0 0 0 0 0] - crit 20890135467565.875

MS 7218      mask [1 1 1 1 0 0 0 0 0 0 0 0 0] - crit 170956927230.7656

MS 7963      mask [1 0 1 1 0 0 0 0 0 0 0 0 0] - crit 170434299057.8078

R 7974 mask [1 1 1 0 0 0 0 0 0 0 0 0 0] - crit 2867674934.0276933

MS 7974      mask [1 1 1 0 0 0 0 0 0 0 0 0 0] - crit 568591638.6494204

R 8108 mask [1 1 0 0 0 0 0 0 0 0 0 0 0] - crit 4759952.773523395

MS 8108      mask [1 1 0 0 0 0 0 0 0 0 0 0 0] - crit 1379438.0483488187

BEST MODEL

R      mask [1 1 0 0 0 0 0 0 0 0 0 0 0] - crit 4759952.773523395

MS     mask [1 1 0 0 0 0 0 0 0 0 0 0 0] - crit 1379438.0483488187

BEST

R test step 2model[ 9.90593871 29.69148175  0.          0.          0.          0.

  0.          0.        ] - crit 0.41043537126313256

MS test step 2      model[ 9.90593871 29.69148175  0.          0.          0.
0.

  0.          0.        ] - crit 0.0926386372078789
```

```
TRAIN SIZE 8
      X train [1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5]
      Y train [18.222 17.216 15.778 14.749 14.014 13.469 13.052 12.724]
TEST SIZE 8
      X test [1. 2. 3. 4. 5. 6. 7. 8.]
      Y test [14.    18.    16.444 15.219 14.352 13.722 13.248 12.879]
Final (the best) regression model:
Y = 9.91/X^0 + 29.69/X^1
Regression crit: 0.41043537126313256
Minimal shift crit: 0.0926386372078789
```



Висновки: було складено програму відшукання моделі оптимальної складності використовуючи однорядний алгоритм самоорганізації моделей, критерій регулярності, критерій мінімуму зсуву (комбінований), а також задано клас опорних функцій.