Національний технічний університет України
«Київський політехнічний інститут ім. І. Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Методи оптимізації та планування експерименту

Лабораторна робота №2

«Проведення двофакторного експерименту з використанням лінійного рівняння регресії»

Виконав:
студент групи ІО-93
Руденко С.О.
Номер залікової книжки:
9327
Перевірив:
ас. Регіда П.Г.

Київ 2021

Лабораторна робота № 2

Тема: «Проведення двофакторного експерименту з використанням лінійного рівняння регресії».
Мета: провести двофакторний експеримент, перевірити однорідність дисперсії за критерієм Романовського, отримати коефіцієнти рівняння регресії, провести натуралізацію рівняння регресії.
Теоретичні основи:

В теорії планування експерименту найважливішою частиною є оцінка результатів вимірів. При цьому використовують апроксимуючі поліноми, за допомогою яких ми можемо описати нашу функцію. В ТПЕ ці поліноми отримали спеціальну назву - регресійні поліноми, а їх знаходження та аналіз - регресійний аналіз. Найчастіше в якості базисної функції використовується ряд Тейлора, який має скінченну кількість членів

$$F(x) = F(a) + \frac{x-a}{1!}F'(a) + \frac{(x-a)^2}{2!}F''(a) + \ldots + \frac{(x-a)^N}{N!}F^{(N)}(a)$$

Але при використанні апроксимуючого полінома Тейлора в його початковому вигляді виникає ряд проблем, пов'язаних із знаходженням похідних, оскільки нам невідома функція, а відомий лише ряд її значень. Тому ми замінюємо поліном Тейлора аналогічним йому рівнянням регресії:

$$\hat{y} = b_0 + \sum_{i=1}^{k} b_i x_i + \sum_{i,j=1}^{k} b_{i,j} x_i x_j + \sum_{i=1}^{k} b_{i,i} x_i^2 + \sum_{i,j,n=1}^{k} b_{i,j,k} x_i x_j x_n + \ldots$$

де k –кількість факторів (кількість x)

Мета даної роботи – дослідити лінійну регресійну модель

$$\hat{y} = b_0 + \sum_{i=1}^{k} b_i x_i$$

1) Результати вимірів вихідної величини у в N експериментах є реалізація нормально розподіленої величини.

2) Дисперсії реалізації в усіх точках факторного простору повинні бути однаковими, оскільки дисперсія не повинна залежати від абсолютного значення величини.

3) Вхідні змінні (фактори) - це незалежні величини, які вимірюються з нескінченно малою похибкою відносно похибки вихідної величини. Будь який багатофакторний експеримент є результатом варіювання усіх факторів.

Варіант 324

324	-30	0	15	50

Приклад роботи програми

(index)	X1	X2	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8
1	-1	-1	-31	43	-26	-19	-28	-40	-31	15
2	1	-1	14	-21	-5	-39	-0	36	37	34
3	-1	1	25	-24	38	5	40	52	12	-14

```
1.0167275481111904 0.8884702171007952 0.9033321454027085
Дисперсія однорідна, оскільки 0.254, 0.357, 0.345 < 2.17
-----Перевірка-----
y1 середнє = -14.625; перевірка =  -14.625
y2 середнє = 7; перевірка =  7.000
y3 середнє = 16.75; перевірка =  16.750
-------------------------
Натуралізоване рівняння:
y=a0+a1x1+a2x2=-6.45+0.72*x1+0.90*x2
-----Перевірка коефіцієнтів натуралізованого рівняння-----
y1 середнє = -14.625; перевірка =  -14.625
y2 середнє = 7; перевірка =  7.000
y3 середнє = 16.75; перевірка =  16.750
```

(index)	X1	X2	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8
1	-1	-1	54	31	45	60	-24	-37	-4	-15
2	1	-1	-6	-34	46	13	13	58	27	48
3	-1	1	49	3	-13	13	-5	49	24	-13

```
Дисперсія однорідна, оскільки 0.160, 0.749, 0.117 < 2.17
-----Перевірка-----
y1 середнє = 13.75; перевірка =  13.750
y2 середнє = 20.625; перевірка =  20.625
y3 середнє = 13.375; перевірка =  13.375
-------------------------
Натуралізоване рівняння:
y=a0+a1x1+a2x2=20.79+0.23*x1+-0.01*x2
-----Перевірка коефіцієнтів натуралізованого рівняння-----
y1 середнє = 13.75; перевірка =  13.750
y2 середнє = 20.625; перевірка =  20.625
y3 середнє = 13.375; перевірка =  13.375
```

(index)	X1	X2	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8
1	-1	-1	5	-15	53	48	-14	56	-34	-8
2	1	-1	6	-22	18	46	-7	43	7	26
3	-1	1	-34	7	44	1	42	-37	44	58

```
Дисперсія однорідна, оскільки 0.770, 0.200, 0.923 < 2.17
-----Перевірка-----
y1 середнє = 11.375; перевірка =  11.375
y2 середнє = 14.625; перевірка =  14.625
y3 середнє = 15.625; перевірка =  15.625
-------------------------
Натуралізоване рівняння:
y=a0+a1x1+a2x2=12.80+0.11*x1+0.12*x2
-----Перевірка коефіцієнтів натуралізованого рівняння-----
y1 середнє = 11.375; перевірка =  11.375
y2 середнє = 14.625; перевірка =  14.625
y3 середнє = 15.625; перевірка =  15.625
```

Текст програми

```
function getRandomInt(min, max) {
    return Math.random() * (max - min) + min;
}

function getMaxOfArray(numArray) {
    return Math.max.apply(null, numArray);
}

function getMinOfArray(numArray) {
    return Math.min.apply(null, numArray);
}

const x1Min = -30
const x1Max = 0
const x2Min = 15
const x2Max = 50

const m = 8

const yMax = (30-24)*10
const yMin = (20-24)*10
```

```
var y1avarage = 0
var y2avarage = 0
var y3avarage = 0

var dispertion1 = 0
var dispertion1 = 0
var dispertion1 = 0



const table_planning = {
    "1": {X1:-1, X2:-1},
    "2": {X1:+1, X2:-1},
    "3": {X1:-1, X2:+1},

}

const exp = () =>{
    for (let i = 0; i<3; i++){
        for (let j = 0; j<m; j++){
            const factor = Math.round(getRandomInt(yMin, yMax)
)
            Object.values(table_planning)[i]["Y"+(j+1)] = fact
or
        }
    }

    console.table(table_planning)

    for (let i = 0; i<3; i++){
        const temp = Object.values(Object.values(table_plannin
g)[i]).slice(2)
        switch(i){
            case 0:
                y1avarage = (temp.reduce((a, b) => a + b, 0))/
m
                dispertion1 = (temp.reduce((a, b) => a + ((b-
y1avarage)**2), 0))/m
                break
```

```
            case 1:
                y2avarage = (temp.reduce((a, b) => a + b, 0))/
m
                dispertion2 = (temp.reduce((a, b) => a + ((b-
y2avarage)**2), 0))/m
                break
            case 2:
                y3avarage = (temp.reduce((a, b) => a + b, 0))/
m
                dispertion3 = (temp.reduce((a, b) => a + ((b-
y3avarage)**2), 0))/m
                break

        }
    }
    const deviation = ((2*((2*m)-2))/(m*(m-4)))**(1/2)

    const Fuv1 = dispertion1 > dispertion2 ? dispertion1/dispe
rtion2 : dispertion2/dispertion1
    const Fuv2 = dispertion3 > dispertion1 ? dispertion3/dispe
rtion1 : dispertion1/dispertion3
    const Fuv3 = dispertion3 > dispertion2 ? dispertion3/dispe
rtion2 : dispertion2/dispertion3

    const Ouv1 = ((m-2)/m)*Fuv1
    const Ouv2 = ((m-2)/m)*Fuv2
    const Ouv3 = ((m-2)/m)*Fuv3

    const Ruv1 = (Math.abs(Ouv1-1)/deviation).toFixed(3)
    const Ruv2 = (Math.abs(Ouv2-1)/deviation).toFixed(3)
    const Ruv3 = (Math.abs(Ouv3-1)/deviation).toFixed(3)

    const criteriaTable  = {
        "0.99": {"2":1.73, "6":2.16, "8":2.43, "10":2.62 , "12
":2.75, "15":2.9},
        "0.98": {"2":1.72, "6":2.13, "8":2.37, "10":2.54 , "12
":2.66, "15":2.8},
        "0.95": {"2":1.71, "6":2.10, "8":2.27, "10":2.41 , "12
":2.52, "15":2.64},
```

```
        "0.90": {"2":1.69, "6":2, "8":2.17, "10":2.49 , "12":2
.39, "15":2.49},

    }

    var Rcr = 0
    for (let i = 3; i>=0; i--){
        let temp = Object.values(criteriaTable)[i][m]
        if (Ruv1<temp && Ruv2<temp && Ruv3<temp){
            Rcr = temp
            console.log(`Дисперсія однорідна, оскі-
льки ${Ruv1}, ${Ruv2}, ${Ruv3} < ${temp}`)
            break
        }
    }
    return Rcr
}

while(true){
    flag = exp()
    if (flag == 0){
        exp()
    }
    else{
        var mx1 = 0;
        var mx2 = 0;
        var my = (y1avarage+y2avarage+y3avarage)/3
        var a1 = 0;
        var a2 = 0
        var a3 = 0;
        for (let i =0; i<2; i++){
            for (let j=0; j<3; j++){
                let temp = Object.values(table_planning)[j]["X
"+(i+1)]
                switch(i){
                    case(0):
                        mx1 += temp/3
                        a1 += ((temp)**2)/3
                        a2 += (temp*Object.values(table_planni
ng)[j]["X"+(i+2)])/3
```

```
                    break
                case(1):
                    mx2 += temp/3
                    a3 += ((temp)**2)/3
                    break
            }
        }
    }

    var a11 = 0
    var a22 = 0

    for (let i =0; i<2; i++){
        for (let j=0; j<3; j++){
            let temp = Object.values(table_planning)[j]["X
"+(i+1)]
            switch(i){
                case(0):
                    if(j == 0){
                        a11 += (temp*y1avarage)/3
                    }
                    if(j == 1){
                        a11 += (temp*y2avarage)/3
                    }
                    if(j == 2){
                        a11 += (temp*y3avarage)/3
                    }
                    break
                case(1):
                    if(j == 0){
                        a22 += (temp*y1avarage)/3
                    }
                    if(j == 1){
                        a22 += (temp*y2avarage)/3
                    }
                    if(j == 2){
                        a22 += (temp*y3avarage)/3
                    }
                    break
            }
```

```
            }
        }

        let matrixDet0 = det(matrix([[my, mx1, mx2], [a11, a1,
 a2], [a22, a2, a3]]))
        let matrixDet1 = det(matrix([[1, my, mx2], [mx1, a11,
a2], [mx2, a22, a3]]))
        let matrixDet2 = det(matrix([[1, mx1, my], [mx1, a1, a
11], [mx2, a2, a22]]))
        let mainDet = det(matrix([[1, mx1, mx2], [mx1, a1, a2]
, [mx2, a2, a3]]))

        let b0 = parseFloat((matrixDet0/mainDet))
        let b1 = parseFloat((matrixDet1/mainDet))
        let b2 = parseFloat((matrixDet2/mainDet))

        console.log("-----Перевірка-----")

        let flags = []
        for (let j=0; j<3; j++){
            let temp1 = Object.values(table_planning)[j]["X1"]
            let temp2 = Object.values(table_planning)[j]["X2"]
            flags.push(b0+(temp1*b1)+(temp2*b2))
        }

        console.log(`y1 середнє = ${y1avarage}; переві-
рка =  ${flags[0].toFixed(3)}`)
        console.log(`y2 середнє = ${y2avarage}; переві-
рка =  ${flags[1].toFixed(3)}`)
        console.log(`y3 середнє = ${y3avarage}; переві-
рка =  ${flags[2].toFixed(3)}`)

        let deltaX1 = (Math.abs(x1Max-x1Min))/2
        let deltaX2 = (Math.abs(x2Max-x2Min))/2

        let x10 = (x1Max+x1Min)/2
        let x20 = (x2Max+x2Min)/2

        a0 = b0-(b1*(x10/deltaX1))-(b2*(x20/deltaX2))
        a1 = b1/deltaX1
```

```
        a2 = b2/deltaX2

        console.log("-------------------------")
        console.log("Натуралізоване рівняння:")
        console.log(`y=a0+a1x1+a2x2=${a0.toFixed(2)}+${a1.toFi
xed(2)}*x1+${a2.toFixed(2)}*x2`)

        console.log("-----Перевірка коефіцієнтів натуралізова-
ного рівняння-----")
        let natur1 = a0+x1Min*a1+x2Min*a2
        console.log(`y1 середнє = ${y1avarage}; переві-
рка =  ${natur1.toFixed(3)}`)

        let natur2 = a0+x1Max*a1+x2Min*a2
        console.log(`y2 середнє = ${y2avarage}; переві-
рка =  ${natur2.toFixed(3)}`)

        let natur3 = a0+x1Min*a1+x2Max*a2
        console.log(`y3 середнє = ${y3avarage}; переві-
рка =  ${natur3.toFixed(3)}`)

        break

    }
}
```

Висновок:

У ході виконання лабораторної роботи ми провели двофакторний експеримент, перевірили однорідність дисперсії за критерієм Романовського, отримали коефіцієнти рівняння регресії, провели натуралізацію рівняння регресії. Були розроблені відповідні тестові програми, що підтверджують правильність виконання завдання. Кінцева мета роботи досягнута.

Відповіді на контрольні питання:

1)Що таке регресійні поліноми і де вони застосовуються?

В теорії планування експерименту найважливішою частиною є оцінка результатів вимірів. При цьому використовують апроксимуючі поліноми, за допомогою яких ми можемо описати нашу функцію. В ТПЕ ці поліноми отримали спеціальну назву - регресійні поліноми, а їх знаходження та аналіз -

регресійний аналіз. Найчастіше в якості базисної функції використовується ряд Тейлора, який має скінченну кількість членів

$$F(x) = F(a) + \frac{x-a}{1!}F'(a) + \frac{(x-a)^2}{2!}F''(a) + \ldots + \frac{(x-a)^N}{N!}F^{(N)}(a)$$

2) Визначення однорідності дисперсії.

Однорідність дисперсії означає, що серед усіх дисперсій нема таких, які б значно перевищували одна одну. Перевірка однорідності проводиться за допомогою різних статистичних критеріїв.

Найпростішим з них є критерій Фішера (F- критерій), який служить для порівняння двох дисперсій та представляє собою відношення більшої дисперсії до меншої $Fp = S_1^2 / S_2^2 > 1$

Якщо $Fp < F$табл для відповідної кількості ступенів свободи та прийнятого рівня значимості, то дисперсії однорідні і навпаки. Для знаходження Fт необхідні два вхідні параметри $\mathbf{f_1}$і $\mathbf{f_2}$: кількості ступенів свободи відповідно для більшої і меншої дисперсій.

3)Що називається повним факторним експериментом?

Повним факторним експериментом (ПФЕ) називається такий експеримент, при реалізації якого визначається значення параметра оптимізації при всіх можливих поєднаннях рівнів варіювання факторів. Якщо ми маємо справу з k факторами, кожен з яких може встановлюватися на q рівнях, то для того, щоб здійснити повний факторний експеримент необхідно поставити $n = q^k$ дослідів.

Найбільшого поширення набули експерименти, в яких фактори варіюються на двох рівнях, тобто експерименти типу 2^k. Планування, проведення та обробка результатів ПФЕ складається з таких етапів: вибір залежних і незалежних змінних (факторів); кодування незалежних (вхідних) чинників; складання план-матриці експерименту; рандомізація дослідів (їх реалізація у випадковому порядку); реалізація плану експерименту; розрахунок і оцінка значимості коефіцієнтів моделі; перевірка адекватності отриманої моделі.