МИНИСТЕРСТВО НАУКИ
И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Новосибирский государственный технический университет»

НГТУ НЭТИ | Факультет прикладной математики и информатики

Кафедра теоретической и прикладной информатики

Лабораторные работы №1 и №2
по дисциплине «Статистические методы анализа данных»

Студенты ИВАНОВ ВЛАДИСЛАВ (92)
ОБЕРШТ ЕЛЕНА (93)

Вариант 5

Преподаватель ПОПОВ АЛЕКСАНДР АЛЕКСАНДРОВИЧ

Новосибирск, 2022

1 Постановка задачи

Произвести моделирование объекта, о котором известно: число факторов – два; по первому фактору зависимость выхода близка к линейной (возрастающей), по второму фактору зависимость близка к параболической, при этом первый фактор в эксперименте может варьироваться на четырех уровнях (принимать только четыре разрешенных значений), а второй на пяти уровнях. Максимальное значение отклика приходится на внутреннюю точку области действия второго фактора.

Спроектировать и сформировать программные модули по вычислению МНК-оценок параметров для заданной параметрической модели объекта. Предусмотреть достаточно простой способ настройки программы на необходимый вид (структуру) модели. Пользуясь полученными экспериментальными данными, оценить параметры модели объекта. Проверить адекватность полученной модели.

2 Описание объекта

2.1 Функция и ограничения

Построим линейную имитационную модель. Поскольку зависимость выхода по первому фактору близка к линейной, выберем досточно большое значение параметра при этом факторе.

Запишем уравнение и зададим области определения для обоих факторов в соответствии с заданными уровнями:

$$\theta = (1, 2, 0.01, -0.5)^T$$

$$x_1 \in \{-1, -0.5, 0.5, 1\}$$

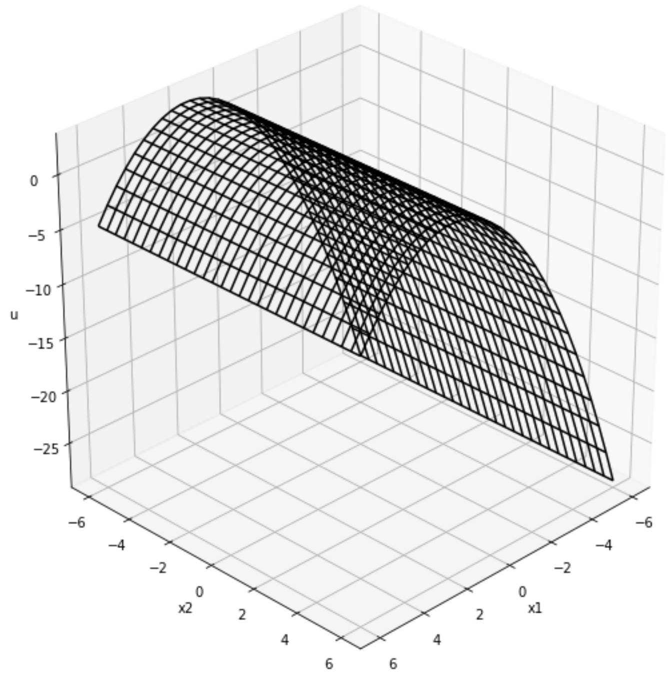
$$x_2 \in \{-1, -0.5, 0, 0.5, 1\}$$

$$u = \eta(x, \theta) = \theta^T f(x_1, x_2) = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_2^2$$

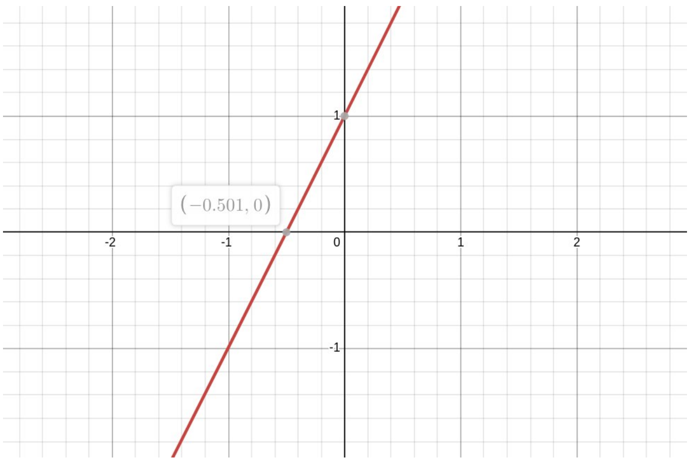
$$= 1 + 2x_1 + 0.01x_1^2 - 0.5x_2^2$$

2.2 Графики зависимости

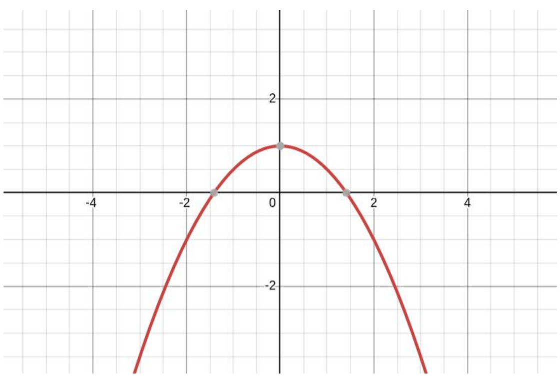
$u(x_1, x_2)$



$u(x_1, 0)$



$u(0, x_2)$



3 Генерация данных

3.1 Параметры

Будем генерировать значения ошибок наблюдений e_j по нормальному закону с нулевым математическим ожиданием и дисперсией $d = 5\%$, а затем прибавлять к истинным значениям u_j. Для нахождения значения стандартного отклонения необходимо найти мощность сигнала ω^2:

$$n = 20$$

$$\rho = 0.05$$

$$\overline{u} = \frac{1}{n}\sum_{j=1}^{n} u_j = 0.76$$

$$\omega^2 = \frac{(u - \overline{u})(u - \overline{u})^T}{n-1} = 2.67$$

$$\sigma = \sqrt{\rho\omega^2} = 0.36$$

$$e_j \sim N(0, \sigma^2)$$

$$y_j = u_j + e_j$$

3.2 Результат

	x1	x2	u	e	y
1	-1.0	-1.0	-1.4900	-0.199549	-1.689549
2	-1.0	-0.5	-1.1150	0.102253	-1.012747
3	-1.0	0.0	-0.9900	-0.087603	-1.077603
4	-1.0	0.5	-1.1150	-0.107579	-1.222579
5	-1.0	1.0	-1.4900	-0.037157	-1.527157
6	-0.5	-1.0	-0.4975	0.375469	-0.122031
7	-0.5	-0.5	-0.1225	0.246374	0.123874
8	-0.5	0.0	0.0025	-0.081125	-0.078625
9	-0.5	0.5	-0.1225	0.012180	-0.110320
10	-0.5	1.0	-0.4975	-0.570820	-1.068320
11	0.5	-1.0	1.5025	-0.212774	1.289726
12	0.5	-0.5	1.8775	-0.109441	1.768059
13	0.5	0.0	2.0025	-0.352896	1.649604
14	0.5	0.5	1.8775	0.372293	2.249793
15	0.5	1.0	1.5025	0.035215	1.537715
16	1.0	-1.0	2.5100	0.573715	3.083715
17	1.0	-0.5	2.8850	0.488241	3.373241
18	1.0	0.0	3.0100	0.209015	3.219015
19	1.0	0.5	2.8850	0.213488	3.098488
20	1.0	1.0	2.5100	0.284083	2.794083

4 Оценка параметров

Метод наименьших квадратов позволяет получить вектор параметров модели, являющийся результатом решения нормального уравнения:

$$\widetilde{\theta} = \arg\min_{\theta}[(y - \theta^T f(x))^T (y - \theta^T f(x))]$$

$$= (X^T X)^{-1} X y$$

Используя новую модель, можно также получить оценку σ:

$$\widetilde{\sigma}^2 = \frac{\widetilde{e}^T \widetilde{e}}{n - m}$$

$$\widetilde{e} = y - \widetilde{y} = y - X\widetilde{\theta}$$

Полученные оценки:

$$\widetilde{\theta} = (1.03, 1.95, 0.05 - 0.41)^T$$

$$\theta = (1, 2, 0.01, -0.5)^T$$

$$\widetilde{\sigma}^2 = 0.12$$

Оценка σ позволяет выполнить проверку гипотезы об адекватности модели:

$$\alpha = 0.05$$

$$f = \infty$$

$$n - m = 16$$

$$F_T = 2.01$$

$$F = \frac{\widetilde{\sigma}^2}{\sigma^2} = 0.95 < F_T$$

Модель является адекватной.

	y	y*	y-y*
1	[-1.6895487564642298]	[-1.5074126750279806]	[-0.18213608143624915]
2	[-1.0127468502812742]	[-1.1272577656614606]	[0.11451091538018643]
3	[-1.077602769204573]	[-1.0005394625392872]	[-0.07706330666528594]
4	[-1.222578518798963]	[-1.1272577656614606]	[-0.09532075313750243]
5	[-1.527156768996937]	[-1.5074126750279806]	[-0.019744093968956378]
6	[-0.12203115297174882]	[-0.6084224688205018]	[0.48639131584875295]
7	[0.12387365029094902]	[-0.22826755945398158]	[0.352141209744930б]
8	[-0.07862541789391711]	[-0.10154925633180834]	[0.022923838437891225]
9	[-0.110320194597582]	[-0.22826755945398158]	[0.11794736485639959]
10	[-1.0683195580528546]	[-0.6084224688205018]	[-0.45989708923235284]
11	[1.2897262691978564]	[1.549444375877988]	[-0.2597181066801315]
12	[1.7680594733750643]	[1.929599285244508]	[-0.16153981186944377]
13	[1.64960437106817]	[2.0563175883666815]	[-0.4067132172985115]
14	[2.249792846883478]	[1.929599285244508]	[0.3201935616389697]
15	[1.5377153104314865]	[1.549444375877988]	[-0.01172906544650143]
16	[3.0837151628468806]	[2.8083210143689987]	[0.2753941484778819]
17	[3.37324072596546]	[3.1884759237355187]	[0.18476480222994152]
18	[3.2190148106007057]	[3.315194226857692]	[-0.0961794162569864 5]
19	[3.0984881039358982]	[3.1884759237355187]	[-0.08998781979962045]
20	[2.794082619545592]	[2.8083210143689987]	[-0.014238394823406608]

```
u_avg: 0.7562499999999999
omega_sq: 2.6776463815789473
sigma: 0.3658993291589196
New theta: [ 1.02727556  1.95398404  0.04760586 -0.41060491]
New sigma^2: 0.1205992702354136
F = 0.9007856382014039
F (SciPy) = 0.9500000000000155
FT = 2.0102414060038662
F < FT: True
```

5 Код программы

```
import pandas as pd
import numpy as np
import random
import scipy.stats
from matplotlib import pyplot as plt

random.seed(42)

def u(x1,x2):
    return 1 + 2*x1 + 0.01*x1**2 - 0.5*x2**2

```

```python
def plot_3d():
    x1 = np.linspace(-6, 6, 30)
    x2 = np.linspace(-6, 6, 30)
    X1, X2 = np.meshgrid(x1, x2)
    U = u(X1, X1)
    fig = plt.figure(figsize=(10,10))
    ax = plt.axes(projection='3d')
    ax.plot_wireframe(X1, X2, U, color='black')
    ax.set_xlabel('x1')
    ax.set_ylabel('x2')
    ax.set_zlabel('u');
    ax.view_init(30, 45)

x1j, x2j, uj = [], [], []
for x1 in [-1,-0.5,0.5,1]:
    for x2 in [-1,-0.5,0,0.5,1]:
        x1j.append(x1)
        x2j.append(x2)
        uj.append(u(x1,x2))

n = 20
p = 0.05
u_avg = np.full((20,), np.mean(uj))
omega_sq = (np.dot((uj-u_avg),(uj-u_avg)))/(n-1)
sigma = np.sqrt(p*omega_sq)
ej = np.random.normal(0, sigma, n)
yj = uj + ej

df = pd.DataFrame(list(zip(x1j, x2j, uj, ej, yj)), columns=['x1', 'x2', 'u', 'e', 'y'])
df.index += 1

x1 = np.ones(n)
x2 = np.reshape(x1j, (n, ))
x3 = pow(np.reshape(x1j, (n, )), 2)
x4 = pow(np.reshape(x2j, (n, )), 2)
X = np.array([x1, x2, x3, x4]).T
y = np.reshape(yj, (n, 1))

theta = np.dot(np.linalg.inv(np.dot(X.T, X)), np.dot(X.T, y))
y_hat = np.dot(X, theta)
e_tilde = y - y_hat
sigma_sq_new = np.dot(e_tilde.T, e_tilde) / (n - theta.shape[0])

F = sigma_sq_new[0][0]/sigma**2
F_scipy = scipy.stats.f.cdf(FT, dfn=9999, dfd=16)
FT = scipy.stats.f.ppf(q=1-0.05, dfn=9999, dfd=16)

print('u_avg:', u_avg[0])
```

```
print('omega_sq:', omega_sq)
print('sigma:', sigma)
print('New theta:', theta.T[0])
print('New sigma^2:', sigma_sq_new[0][0])
print('F =', F)
print('F (SciPy) =', F_scipy)
print('FT =', FT)
print('F < FT:', F < FT)
```