

Отчёт №8 (вар.14)

Оценка квадратичной и
линейной функции
методом наименьших
квадратов

Романенко Демьян, М3238

`romanenko@niuitmo.ru`

18.06.2020

1 Постановка задачи

1. Квадратичная функция

- Смоделировать квадратичную функцию $y(x) = \alpha_1 x^2 + \alpha_2 x + \alpha_3$, наблюдаемую в нормальных шумах в пакете Octave в соответствии с параметрами варианта;
- Оценить коэффициенты квадратичной зависимости, уровень шумов и квадратичную функцию по зашумленным данным;
- Сравнить полученные результаты с "истинными" данными.

2. Линейная функция

- Смоделировать линейную функцию $y(x) = c_1 x + c_2$, наблюдаемую в нормальных шумах в пакете Octave в соответствии с параметрами варианта;
- Оценить коэффициенты линейной зависимости, уровень шумов и линейную функцию по зашумленным данным;
- Построить доверительный интервал для значений функции при уровне доверия 0.95;
- Сравнить полученные результаты с "истинными" данными.

Исходные данные

1. $x_{min} = -2.2$, $x_{max} = 2.5$ — границы интервала;
2. $n = 69$ — число точек;
3. $a_1 = 1.7$, $a_2 = -2.4$, $a_3 = -3.6$ — коэффициенты квадратичной функции;
4. $c_1 = 3.5$, $c_2 = -4.2$ — коэффициенты линейной функции;
5. $s = 1.5$ — уровень шумов.

2 Исходный код программ

2.1 Квадратичная функция

Listing 1: quad.m

```
1  pkg load statistics;  
2  
3  clc;  
4  clear;  
5  
6  x_min = -2.2;  
7  x_max = 2.5;  
8  n = 69;
```

```

9  a = [1.7, -2.4, -3.6];
10 s = 1.5;
11
12 X = (x_min : (x_max - x_min) / (n - 1) : x_max)';
13 y = polyval(a, X);
14
15 Z = s * randn(n, 1);
16 Y = y + Z;
17 plot(X, y, X, Y, '+');
18
19 m = 2;
20 an = polyfit(X, Y, m);
21 Yn = polyval(an, X);
22 plot(X, Y, '+', X, y, X, Yn, 'o');
23
24 e = Yn - Y;
25 printf("Actual coefficients a1 = %d, a2 = %d, a3 = %d;\n", a);
26 printf("Approximated coefficients: %d, %d, %d;\n", an);
27 printf("Scalar product e and Yn: %d;\n", e' * Yn);
28 printf("Noise(s) = %d; Noise(sn) = %d.\n", s, sqrt(e' * e / (n - 2)));

```

2.2 Линейная функция

Listing 2: linear.m

```

1  pkg load statistics;
2
3  clc;
4  clear;
5
6  x_min = -2.2;
7  x_max = 2.5;
8  n = 69;
9  c = [3.5, -4.2];
10 s = 1.5;
11
12 X = (x_min : (x_max - x_min) / (n - 1) : x_max)';
13 y = polyval(c, X);
14

```

```

15 Z = s * randn(n,1);
16 Y = y + Z;
17 plot(X, y, X, Y);
18
19 % linear regression
20 xn = mean(X);
21 yn = mean(Y);
22 cov = (X - xn)' * (Y - yn) / (n - 1);
23 b = cov / (std(X) ^ 2);
24 Yn = yn + b * (X - xn);
25
26 % matlab interpolation
27 m = 1;
28 cn = polyfit(X, Y, m);
29 Ynn = polyval(cn, X);
30 plot(X, Y, '+', X, y, X, Yn, X, Ynn, 'o');
31
32 e = Yn - Y;
33 sn = sqrt(e' * e / (n - 2));
34
35 ta = 1.96;
36 ha = ta * (sn / sqrt(n));
37 da = ha * (1 + (X - xn) .^ 2 / (std(X) ^ 2)) .^ (1 / 2);
38 Yn1 = Yn - da;
39 Yn2 = Yn + da;
40 plot(X, Yn1, X, Yn2, X, Y, 'o', X, Yn)
41
42 printf("Actual coefficients c1 = %d, c2 = %d;\n", c);
43 printf("Approximated coefficients: %d, %d;\n", cn);
44 printf("Matlab coefficients: %d, %d;\n", b, yn - b * xn);
45 printf("Scalar product e and Yn: %d;\n", e' * Yn);
46 printf("Noise(s) = %d; Noise(sn) = %d.\n", s, sn);

```

3 Результаты работы программ

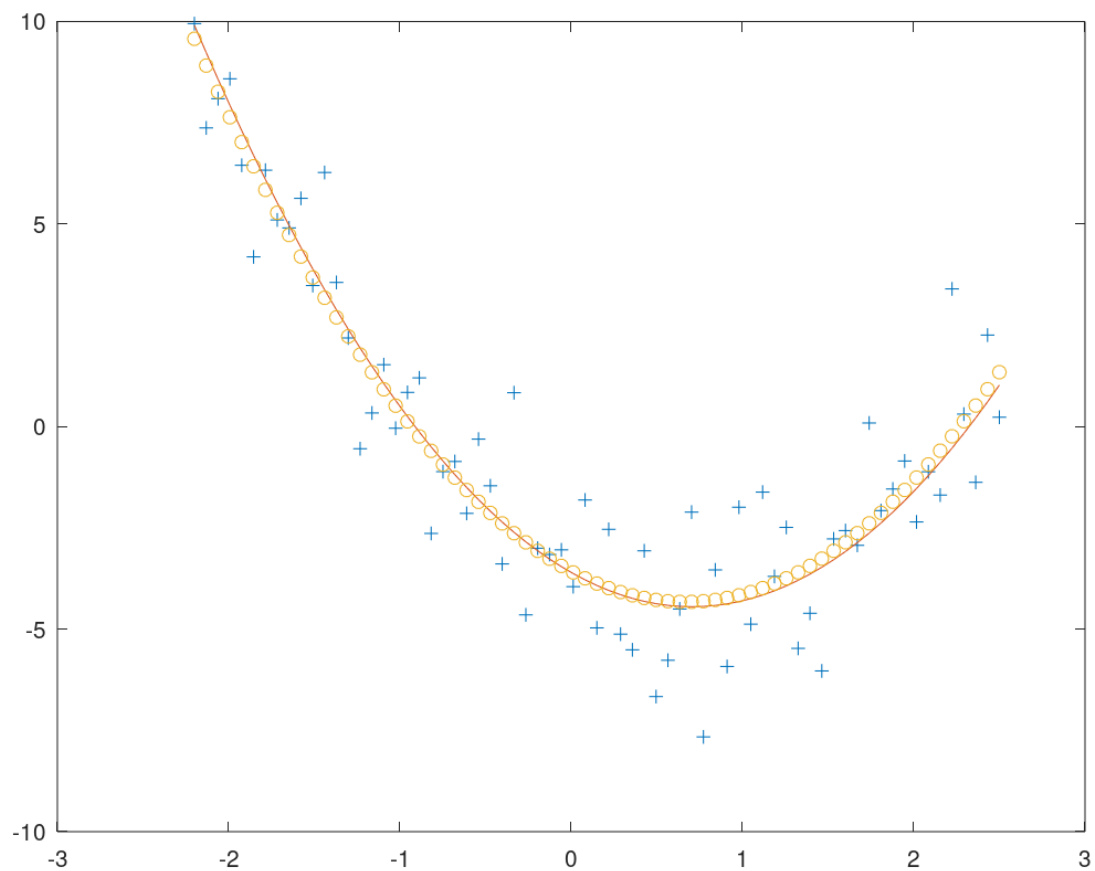
3.1 Квадратичная функция

Actual coefficients $a_1 = 1.7$, $a_2 = -2.4$, $a_3 = -3.6$;

Approximated coefficients: 1.69016, -2.25855, -3.57421;

Scalar product e and Y_n : $3.57492e-14$;

Noise(s) = 1.5; Noise(sn) = 1.48824.



3.2 Линейная функция

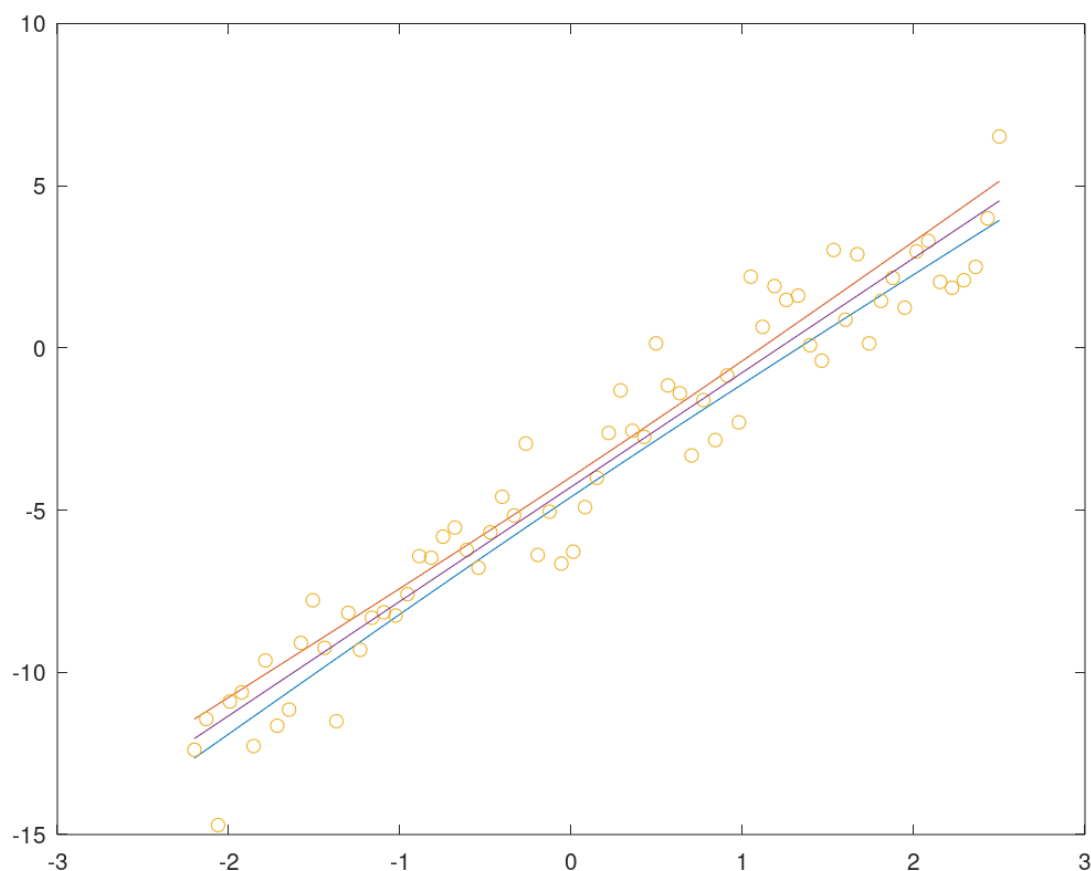
Actual coefficients $c_1 = 3.5$, $c_2 = -4.2$;

Approximated coefficients: 3.52776, -4.28172;

Matlab coefficients: 3.52776, -4.28172;

Scalar product e and Y_n : $4.70735e-13$;

Noise(s) = 1.5; Noise(sn) = 1.29854.



4 Вывод

Полученная функция попадает в доверительный интервал. Для обеих функций посчитанные коэффициенты близки к исходным. Вектор несвязок практически ортогонален вектору значений зашумлённой функции. Посчитанный уровень шумов близок к теоретическому.