Университет ИТМО

Кафедра Высшей Математики

Лабораторная работа №7

«Оценка функции регрессии методом наименьших квадратов»

По дисциплине «Математическая статистика»

Вариант 14

Выполнили:

Стоянов Дмитрий, группа R3496

Замиралов Александр, группа R3496

Самарин Антон, группа R3495

Преподаватель:

Суслина Ирина Александровна

Дата: 10.11.2018

Задача

1. Смоделировать квадратичную функцию, наблюдаемую в нормальных шумах в соответствии с параметрами варианта. Оценить коэффициенты квадратичной зависимости, уровень шумов и квадратичную функцию по зашумленным данным. Сравнить полученные результаты с «истинными данными»
2. Смоделировать линейную функцию, наблюдаемую в нормальных шумах в соответствии с параметрами варианта. Оценить коэффициенты линейной зависимости, уровень шумов и линейную функцию по зашумлённым данным. Сравнить полученные результаты с «истинными» данными
3. Проверить ортогональность вектора , где – зашумленные данные, – восстановленная функция или , или c базисом или

Входные данные:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Xmin | Xmax | n | A1 | A2 | A3 | C1 | C2 | S |
| -2.2 | 2.5 | 69 | 1.7 | -2.4 | -3.6 | 3.5 | -4.2 | 1.5 |

Ход выполнения

**Квадратичная регрессия**

Рассмотрим случай квадратичной регрессии. Добавим шумы в «истинную» неизвестную функцию. Далее восстановим исходную функцию матричным способом и используя готовые функции математического пакета Octave.

clc

clear

x\_min = -2.2;

x\_max = 2.5;

n = 70;

X = (x\_min: (x\_max - x\_min) / (n - 1) : x\_max);

a1 = 1.7;

a2 = -2.4;

a3 = -3.6;

a = [a3, a2, a1]

# unknown function

y = a1 + a2 \* X + a3 \* X .^ 2;

# noise

s = 1.5;

Z = s \* normrnd(0, 1, 1, n);

Y = y + Z;

# recovery of function y(x)

# matrix way

A = [ones(1, n); X; X .^ 2]';

B = A' \* A;

C = A' \* Y';

an = B^(-1) \* C;

an1 = flip(an')

Yn = A \* an;

# using octave functions

m = 2;

a11 = polyfit(X, Y, m)

Yn1 = polyval(a11, X);

plot(X, Y, 'b', X, y, 'r', X, Yn, 'g\*--', X, Yn1, 'po')

legend('Function with noise', 'Unknown function',

'Linear regression (matrix way)','Linear regression (using octave funcs)')

r = Yn - Y';

ort = r' \* Yn

ssn = 1.5;

sn = sqrt(r' \* r / (n-3));

ss1 = [ssn, sn]

**Вывод программы:**

a =

-3.6000 -2.4000 1.7000

an1 =

-3.4865 -2.4444 1.3622

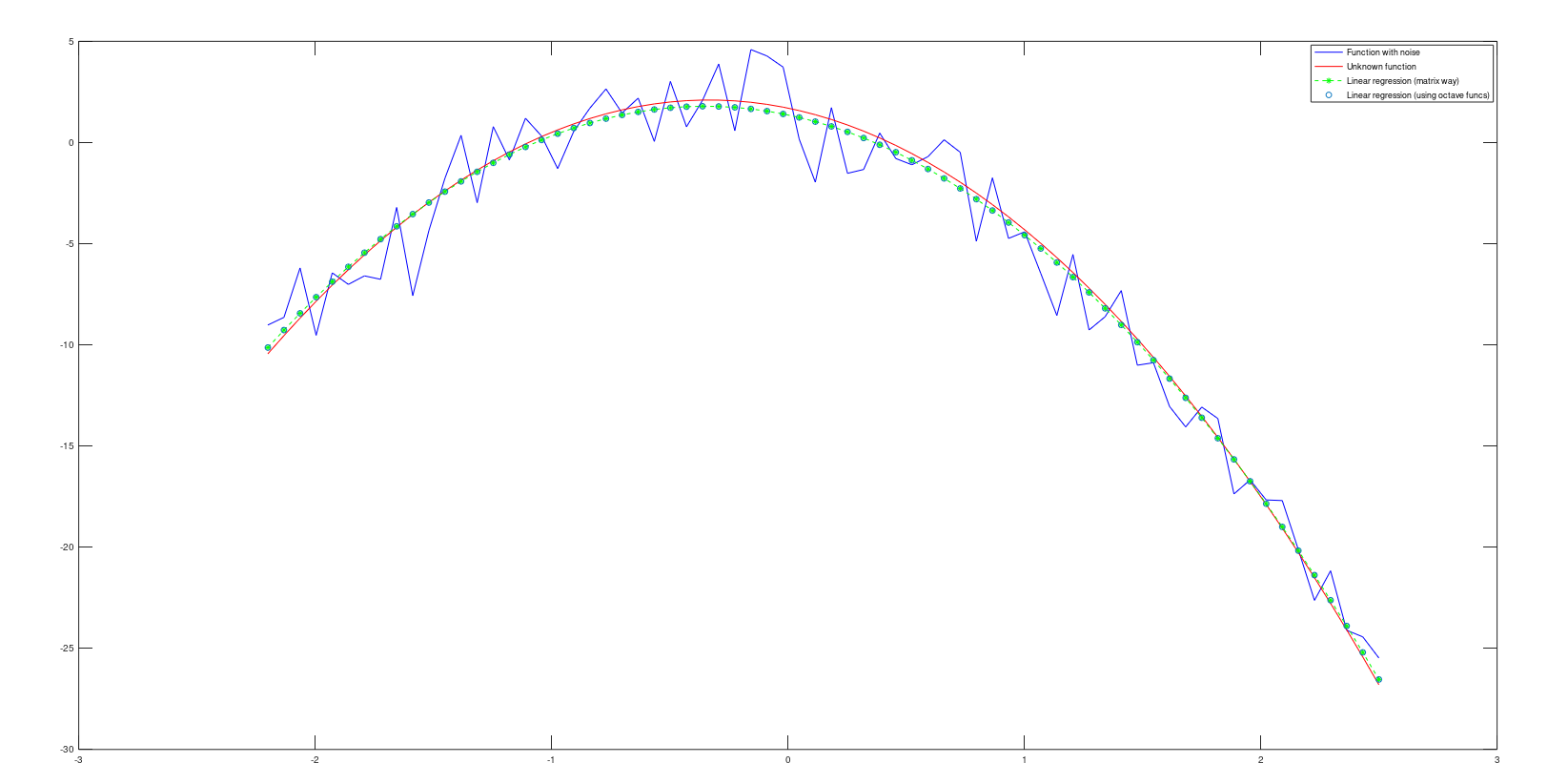
a11 =

-3.4865 -2.4444 1.3622

ort = -1.7053e-12

ss1 =

1.5000 1.5106



**Линейная регрессия**

Восстановим исходную функцию с помощью матричного метода, через корреляцию и с помощью функций математического пакета Octave

clc

clear

x\_min = -2.2;

x\_max = 2.5;

n = 70;

X = x\_min: (x\_max - x\_min) / (n - 1) : x\_max;

c1 = 2.8;

c2 = -3.7;

c = [c2, c1]

# unknown function

y = c1 + c2 \* X;

# noise

s = 1.5;

Z = s \* normrnd(0, 1, 1, n);

Y = y + Z;

# recovery of function y(x)

# matrix way

A = [ones(1, n); X]';

B = A' \* A;

C = A' \* Y';

cn = B^(-1) \* C;

cn\_matrix = flip(cn')

Yn1 = A \* cn;

# cov way

xn = mean(X);

yn = mean(Y);

K = (X - xn)' \* (Y - yn) / (n - 1);

b = K / (std(X) ^ 2);

Yn2 = yn + b \* (X-xn)';

# using octave functions

m = 1;

cn\_octave = polyfit(X, Y, m)

Yn3 = polyval(cn\_octave, X);

plot(X, Y, 'b', X, y, 'r', X, Yn1, 'g\*--', X, Yn2, 'ko', X, Yn3, 'b.')

legend('Function with noise', 'Unknown function',

'Linear regression (matrix way)', 'Linear regression (correlation way)',

'Linear regression (using octave funcs)')

r = Yn1 - Y';

ort = r' \* Yn1

sn = sqrt(r' \* r / (n - 2));

ss = [s, sn]

**Вывод программы:**

c =

-3.7000 2.8000

cn\_matrix =

-3.6906 3.0967

cn\_cov =

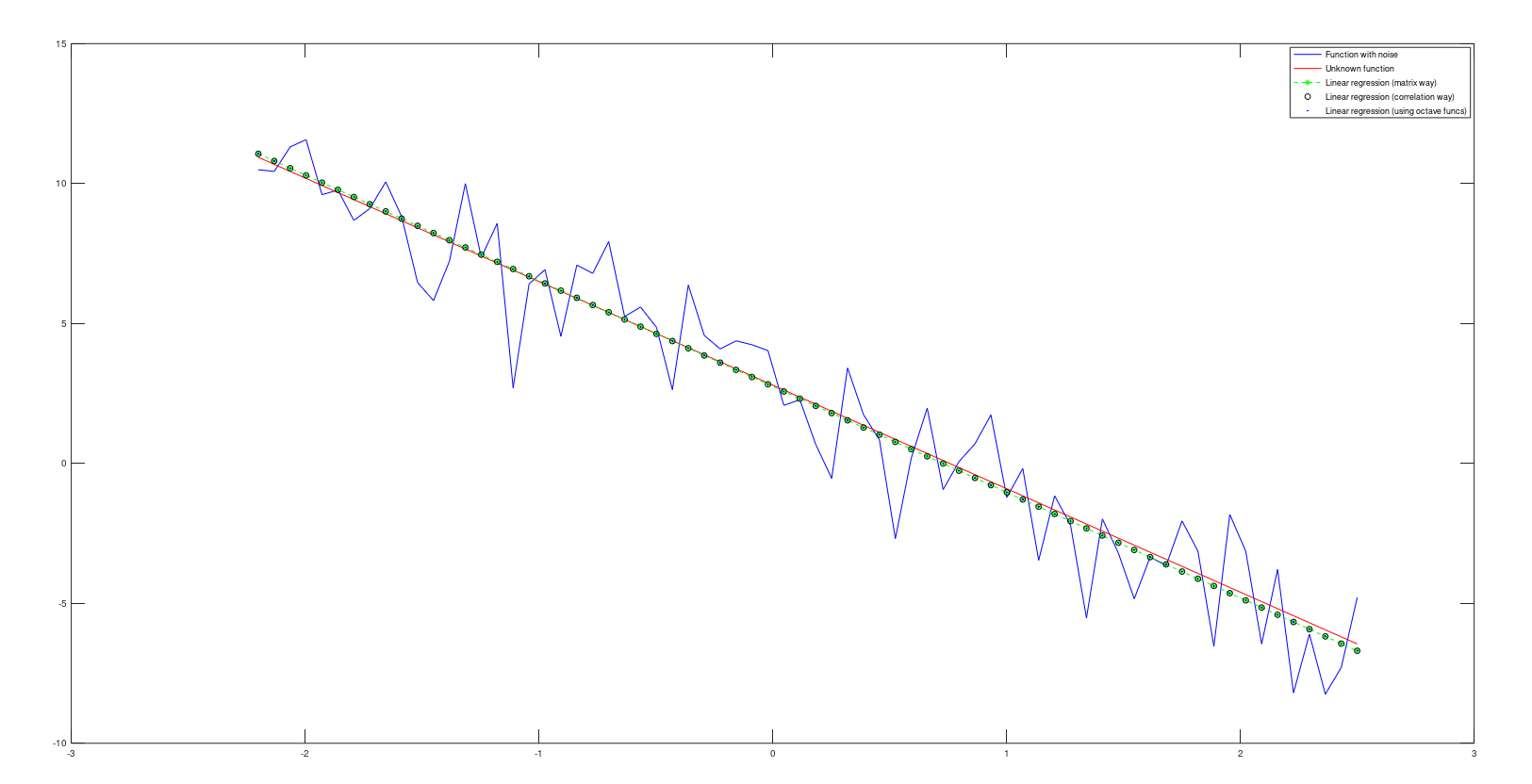
-3.6906 3.0967

cn\_octave =

-3.6906 3.0967

ort = 6.8034e-13

ss = 1.5000 1.3872



Вывод: