ТВМС, Лабораторная работа 8, Вариант 10

Ковешников Глеб, M3238
kovg16@gmail.com

18 июня 2020 г.

Формулировка

1.1 Задание 1

- Смоделировать квадратичную функцию, наблюдаемую в нормальных шумах в соответствии с заданными параметрами.
- Оценить коэффициенты квадратичной зависимости, уровень шумов и квадратичную функцию по зашумленным данным.
- Сравнить результаты с исходными данными.

1.2 Задание 2

- Смоделировать линейную функцию, наблюдаемую в нормальных шумах в соответствии с заданными параметрами.
- Оценить коэффициенты линейной зависимости, уровень шумов и линейную функцию по зашумленным данным.
- Построить доверительный интервал для значений функции при уровне доверия 0.95.
- Сравнить результаты с исходными данными.

Входные данные

- Границы интервала: $x_{min} = 1.3, x_{max} = 2.5$
- Число точек: $n = 60$
- Коэффициенты квадратичной функции: $a_1 = 2.2, a_2 = -3.2, a_3 = 4.5$
- Коэффициенты линейной функции: $c_1 = 2.2, c_2 = 1.5$
- Уровень шумов: $s = 3.7$

Исходный код

3.1 Квадратичная функция

```
xmin = 1.3;
xmax = 2.5;
n = 60;
a1 = 2.2;
a2 = -3.2;
a3 = 4.5;
s = 3.7;

X = (xmin : (xmax - xmin) / (n - 1) : xmax)';
Ys = a3 + a2 * X + a1 * X.^2;
Z = s * randn(n, 1);
Y = Z + Ys;
plot(X, Ys, X, Y, '+');
```

```
m = 2;
an = polyfit(X, Y, m);
Yn = polyval(an, X);
plot(X, Y,'+', X, Ys, X, Yn,'o')

diff = Yn - Y;
sProd = diff' * Yn;
sn = sqrt(diff' * diff / (n - 2));

printf("Real␣coefficients:␣%d,␣%d,␣%d\n", a1, a2, a3);
printf("Interpolated␣coefficients:␣%d,␣%d,␣%d\n", an(1), an(2), an(3));
printf("Scalar␣product␣:␣%d\n", sProd);
printf("Real␣noise␣level␣:␣%d\n", s);
printf("Calculated␣noise␣level␣:␣%d\n", sn);
```

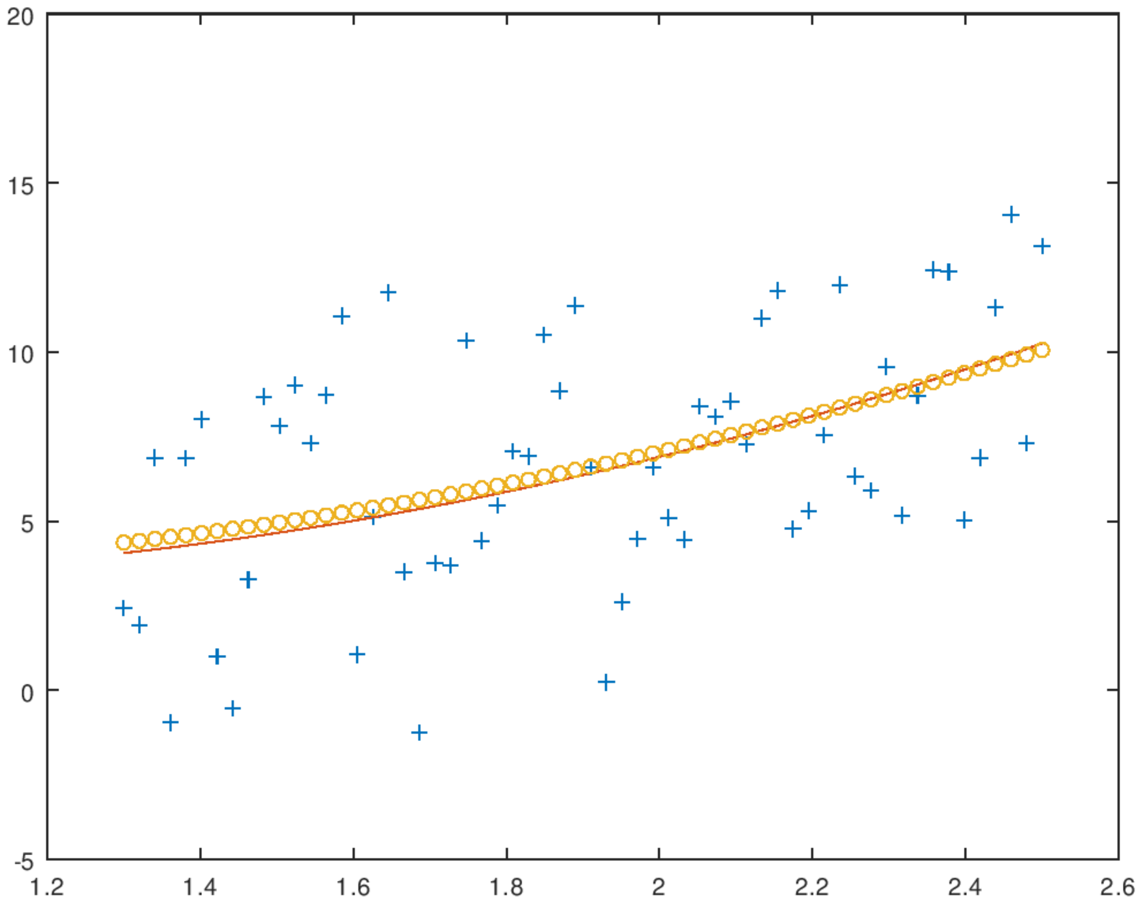
3.2 Линейная функция

```
xmin = 1.3;
xmax = 2.5;
n = 60;
c1 = 2.2;
c2 = 1.5;
s = 3.7;

X = (xmin : (xmax - xmin) / (n - 1) : xmax)';
Ys = c2 + c1 * X;
Z = s * randn(n, 1);
Y = Z + Ys;
plot(X, Ys, X, Y, '+');

xn = mean(X);
yn = mean(Y);
cov = (X - xn) '*␣(Y␣-␣yn)␣/␣(n␣-␣1);
b␣=␣cov␣/␣(std(X)^2);
a␣=␣yn␣-␣b␣*␣xn;
Yn␣=␣yn␣+␣b␣*␣(X␣-␣xn);

m␣=␣1;
cn␣=␣polyfit(X,␣Y,␣m);
Ynn␣=␣polyval(cn,␣X);
plot(X,␣Y,'+',␣X,␣Ys,␣X,␣Yn,␣X,␣Ynn,'o');

diff = Yn - Y;
sProd = diff' * Yn;
sn = sqrt(diff' * diff / (n - 2));

ta = 1.96;
ha = ta * (sn / sqrt(n));
da = ha * (1 + (X - xn).^2 / (std(X)^2)).^(1/2);
Yn1 = Yn - da;
Yn2 = Yn + da;
plot(X, Yn1, X, Yn2, X, Y,'o',X, Yn, X, Ys,'*');

printf("Real␣coefficients:␣%d,␣%d\n", c1, c2);
printf("Interpolated␣coefficients:␣%d,␣%d\n", cn(1), cn(2));
printf("Interpolated␣coefficients␣with␣Matlab:␣%d,␣%d\n", b , a);
printf("Scalar␣product:␣%d\n", sProd);
printf("Read␣noise␣level:␣%d\n", s);
printf("Calculated␣noise␣level:␣%d\n", sn);
```

Результат работы программ

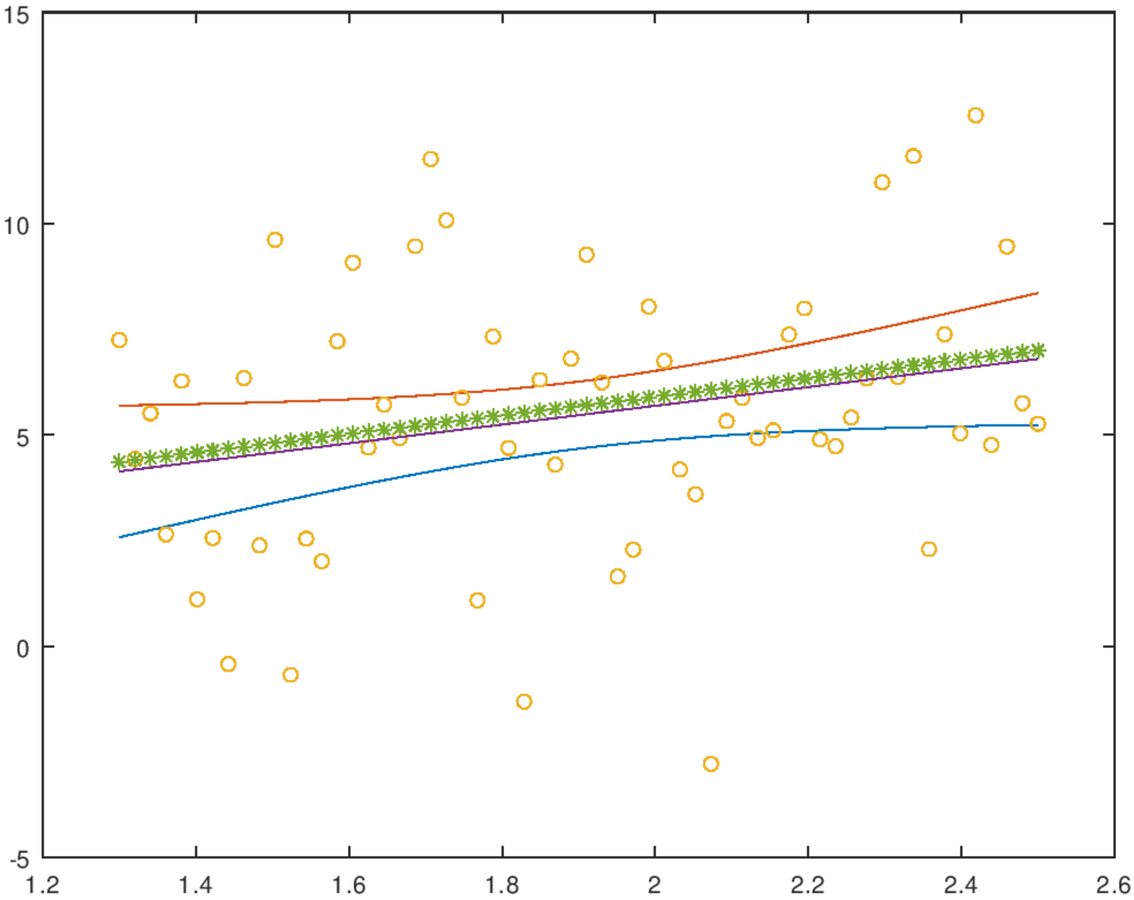
4.1 Квадратичная функция

```
Real coefficients: 2.2, -3.2, 4.5
Interpolated coefficients: 1.8132, -2.1471, 4.09307
Scalar product : 1.7053e-13
Real noise level : 3.7
Calculated noise level : 3.30297
```



4.2 Линейная функция

```
Real coefficients: 2.2, 1.5
Interpolated coefficients: 2.94447, 0.248198
Interpolated coefficients with Matlab: 2.94447, 0.248198
Scalar product: 1.11822e-12
Read noise level: 3.7
Calculated noise level: 3.91222
```



Вывод

Для обеих функций:

- Полученная в результате интерполяции функция хорошо приближается исходной.
- Посчитанный уровень шумов близок к теоретическому.
- Вектор несвязок практически ортогонален вектору значений зашумленной функии.

А также график линейной функций с посчитанными коэффициентами попадает в доверительный интервал.