# Министерство науки и высшего образования Российской Федерации

## Федеральное государственное бюджетное образовательное учреждение высшего образования

## «Новосибирский государственный технический университет»



## Кафедра теоретической и прикладной информатики

### Лабораторная работа №2 по дисциплине «Основы теории машинного обучения»

**УСТОЙЧИВЫЕ МЕТОДЫ ОЦЕНИВАНИЯ ПАРАМЕТРОВ РЕГРЕССИИ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Факультет: | ПМИ |  |  |
| Группа: | ПМИМ-01 |  |  |
| Студенты: | Ершов П.К.  Малышкина Е.Д.  Слободчикова А.Э. | | |
| Вариант: | 4 |  |  |
| Преподаватель: | Попов А.А. |  |  |
|  |  |  |  |  |

Новосибирск

2021

1. **Цель.**

Разработать программу, реализующую поиск М-оценок параметров

итерационным МНК.

1. **Содержание работы.**
2. Изучить методы устойчивого оценивания параметров регрессии и метод поиска значений оценок.
3. Выбрать полиномиальную невысокого порядка (квадратичную, кубическую) модель зависимости отклика y от одного фактора x. По данной модели сгенерировать экспериментальные данные, содержащие выбросы. Выбросы можно смоделировать, увеличив в несколько раз (в несколько десятков раз) величину ошибки в нескольких точках выборки. Проконтролировать наличие выбросов в выборке визуально. Сформировать несколько выборок с различной степенью засорения. Степень засорения варьировать от 1 до 25% с шагом в 4-5%.
4. Разработать программу, реализующую поиск М-оценок параметров итерационным МНК или по методу псевдонаблюдений для функции потерь, указанной в варианте.
5. Выбрать несколько значений параметра функции потерь (включая указанные в варианте) и найти значения М-оценок для каждого из них. В качестве начального приближения оценок в алгоритме итерационного МНК можно взять истинные значения параметров или можно произвести поиск глобального оптимума путем многократного применения итерационного МНК из различных начальных приближений. Вычислить МНК-оценку. Сравнить качество всех полученных оценок по величинам , , где - незашумленный выход объекта. Выбрать значение параметра функции потерь, дающее наилучшее качество при заданной степени засорения.
6. Оформить отчет, включающий в себя

– постановку задачи; – полученный набор данных и значения ошибок наблюдений;

– оценки параметров и результаты их сравнения;

1. для наилучшей из оценок: значения весов наблюдений w на последней итерации, значения , оценку параметра σ;

– графики, отражающие качество оценивания (один из двух вариантов):

1. график зависимостей от фактора x измеренных значений y и прогнозных значений для МНК-оценки и для наилучшей из Моценок (пример графика см. в конце п. 3.1).
2. два графика – график наблюдений и график зависимостей от фактора x истинных значений отклика и прогнозных значений для МНКоценки и для наилучшей из М-оценок

– текст программы.

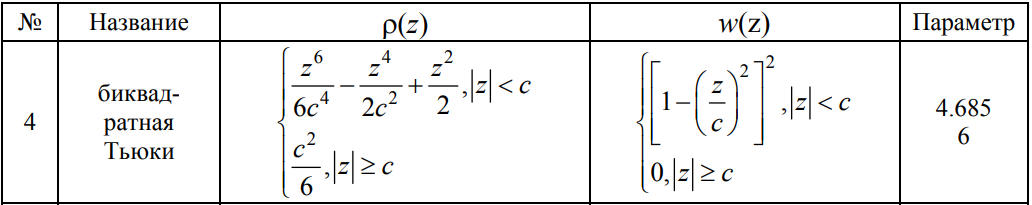
1. **Исходные данные**.

Дана модель:

, где , а e = 5.

Значит .

Вариант задания:



1. **Исследование**:

Сгенерируем наблюдения для разных уровней зашумлённости (выбросы выделены оранжевым цветом):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| % Шума | 0% | 5% | 10% | 15% | 20% | 25% |
| Наблюдения | 15 | 15,458 | 15,458 | 15,458 | 15,458 | 15,458 |
|  | 12,286 | 11,985 | 11,985 | 11,985 | 11,985 | 11,985 |
|  | 10,067 | 10,648 | 10,648 | 10,648 | 10,648 | 10,648 |
|  | 8,299 | 7,998 | 7,998 | 7,998 | 0,781 | 0,781 |
|  | 6,941 | 6,143 | 6,143 | 6,143 | 6,143 | 6,143 |
|  | 5,953 | 6,152 | 6,152 | 6,152 | 6,152 | 6,152 |
|  | 5,292 | 5,895 | 20,365 | 20,365 | 20,365 | 20,365 |
|  | 4,917 | 5,445 | 5,445 | 5,445 | 5,445 | 5,445 |
|  | 4,787 | 5,213 | 5,213 | 5,213 | 5,213 | 5,213 |
|  | 4,859 | 5,204 | 5,204 | 5,204 | 5,204 | 5,204 |
|  | 5,093 | 4,981 | 4,981 | 4,981 | 2,284 | 2,284 |
|  | 5,447 | 5,186 | 5,186 | 5,186 | 5,186 | -1,08 |
|  | 5,88 | 5,46 | 5,46 | 5,46 | 5,46 | 5,46 |
|  | 6,349 | 6,939 | 6,939 | 6,939 | 6,939 | 6,939 |
|  | 6,814 | 6,149 | 6,149 | 6,149 | 6,149 | 6,149 |
|  | 7,232 | 7,17 | 7,17 | 7,17 | 7,17 | 5,677 |
|  | 7,563 | 7,314 | 7,314 | 7,314 | 7,314 | 7,314 |
|  | 7,765 | 6,791 | 6,791 | 6,791 | 6,791 | 6,791 |
|  | 7,796 | 6,872 | 6,872 | 6,872 | 6,872 | 6,872 |
|  | 7,615 | 7,747 | 7,747 | 10,923 | 10,923 | 10,923 |
|  | 7,18 | 7,647 | 7,647 | 7,647 | 7,647 | 7,647 |
|  | 6,45 | 5,865 | 5,865 | 5,865 | 5,865 | 5,865 |
|  | 5,383 | 14,628 | 14,628 | 14,628 | 14,628 | 14,628 |
|  | 3,938 | 3,382 | 3,382 | 3,382 | 3,382 | 3,382 |
|  | 2,073 | 1,082 | 1,082 | 1,082 | 1,082 | 1,082 |
|  | -0,253 | -0,85 | -0,85 | -0,85 | -0,85 | -0,85 |
|  | -3,082 | -0,974 | -0,974 | -0,974 | -0,974 | -0,974 |
|  | -6,455 | -7,504 | -7,504 | -7,504 | -7,504 | -7,504 |
|  | -10,414 | -10,404 | -10,404 | -10,404 | -10,404 | -10,404 |
|  | -15 | -14,849 | -14,849 | -14,849 | -14,849 | -14,849 |

В качестве алгоритма оценки параметров функции потерь будем использовать итерационный МНК.

Для нахождения наилучших проварьируем параметр c : [3.37, 4.685, 6, 7.315, 8.63, 9.945].

Засорение 5%.

|  |  |  |  |
| --- | --- | --- | --- |
| **w** | **y** |  |  |
| 1 | 15.458 | 15.475 | 1.741e-02 |
| 1 | 11.985 | 12.578 | 0.5931 |
| 1 | 10.648 | 10.209 | -0.4390 |
| 1 | 7.998 | 8.324 | 0.3256 |
| 1 | 6.143 | 6.879 | 0.7361 |
| 1 | 6.152 | 5.830 | -0.3221 |
| 1 | 5.895 | 5.133 | -0.7621 |
| 1 | 5.445 | 4.744 | -0.7013 |
| 1 | 5.213 | 4.619 | -0.5938 |
| 1 | 5.204 | 4.715 | -0.4895 |
| 1 | 4.981 | 4.986 | 4.891e-03 |
| 1 | 5.186 | 5.389 | 0.2030 |
| 1 | 5.460 | 5.881 | 0.4209 |
| 1 | 6.939 | 6.416 | -0.5235 |
| 1 | 6.149 | 6.951 | 0.8018 |
| 1 | 7.170 | 7.442 | 0.2723 |
| 1 | 7.314 | 7.845 | 0.5311 |
| 0.999 | 6.791 | 8.116 | 0.1325 |
| 0.999 | 6.872 | 8.211 | 0.1340 |
| 1 | 7.747 | 8.086 | 0.3393 |
| 1 | 7.647 | 7.697 | 4.990e-02 |
| 1 | 5.865 | 7.000 | 0.1135 |
| 0.9015 | 14.628 | 5.950 | -0.8678 |
| 1 | 3.382 | 4.505 | 0.1123 |
| 0.999 | 1.082 | 2.619 | 0.1536 |
| 1 | -0.850 | 0.249 | 0.1098 |
| 0.999 | -0.974 | -2.649 | -0.1676 |
| 0.999 | -7.504 | -6.120 | 0.1384 |
| 1 | -10.404 | -10.206 | 0.1977 |
| 1 | -14.849 | -14.953 | -0.1044 |

|  |  |  |  |
| --- | --- | --- | --- |
| c |  | MSE |  |
| 3.37 | 4.86633029 0.59015686 0.27677857 -0.05341439 | 1.37679e-01 | 2.6724602e-02 |
| 4.685 | 4.87862788 0.57651741 0.27411963 -0.05300906 | 1.08046e-01 | 2.1176916e-02 |
| 6 | 4.87452259 0.58107969 0.27500794 -0.05314471 | 1.17509e-01 | 2.2953783e-02 |
| 7.315 | 4.86472058 0.5919422 0.27712661 -0.05346744 | 1.41839e-01 | 2.7501765e-02 |
| 8.63 | 4.86448746 0.59220089 0.27717702 -0.05347513 | 1.42447e-01 | 2.7615320e-02 |
| 9.945 | 4.87165411 0.58425212 0.27562747 -0.05323891 | 1.24384e-01 | 2.4238345e-02 |

-0,38744.

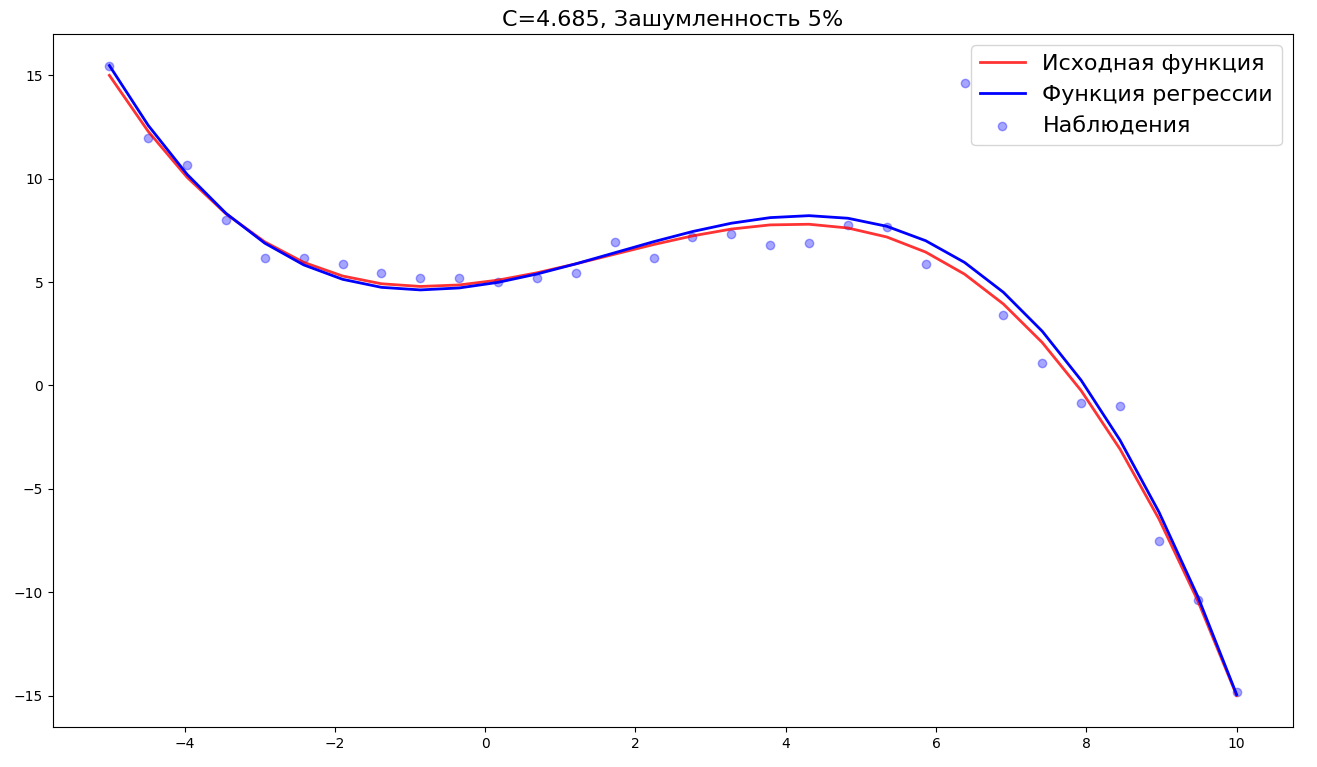


Рисунок 1. Графики наблюдений, функции регрессии и исходной функции для наилучшего значения M-оценки

Засорение 10%.

|  |  |  |  |
| --- | --- | --- | --- |
| **w** | **y** |  |  |
| 1 | 15.458 | 15.642 | 0.1841 |
| 1 | 11.985 | 13.098 | 1.113 |
| 1 | 10.648 | 11.012 | 0.3642 |
| 1 | 7.998 | 9.344 | 1.346 |
| 1 | 6.143 | 8.057 | 1.914 |
| 1 | 6.152 | 7.110 | 0.9581 |
| 0.9153 | 20.365 | 6.465 | -13.90 |
| 1 | 5.445 | 6.084 | 0.6388 |
| 1 | 5.213 | 5.927 | 0.7143 |
| 1 | 5.204 | 5.956 | 0.7523 |
| 1 | 4.981 | 6.132 | 1.151 |
| 1 | 5.186 | 6.416 | 1.230 |
| 1 | 5.460 | 6.769 | 1.309 |
| 1 | 6.939 | 7.152 | 0.2128 |
| 1 | 6.149 | 7.527 | 1.378 |
| 1 | 7.170 | 7.854 | 0.6842 |
| 1 | 7.314 | 8.095 | 0.7810 |
| 1 | 6.791 | 8.211 | 1.420 |
| 1 | 6.872 | 8.164 | 1.292 |
| 1 | 7.747 | 7.913 | 0.1662 |
| 1 | 7.647 | 7.421 | -0.2259 |
| 1 | 5.865 | 6.649 | 0.7840 |
| 0.9841 | 14.628 | 5.557 | -9.071 |
| 1 | 3.382 | 4.108 | 0.7256 |
| 1 | 1.082 | 2.261 | 1.179 |
| 1 | -0.850 | -0.021 | 0.8284 |
| 1 | -0.974 | -2.778 | -1.804 |
| 1 | -7.504 | -6.048 | 1.456 |
| 1 | -10.404 | -9.871 | 0.5331 |
| 1 | -14.849 | -14.285 | 0.5643 |

|  |  |  |  |
| --- | --- | --- | --- |
| С |  | MSE |  |
| 3.37 | 6.15707123 0.36824809 0.22226842 -0.04629675 | 6.67319e-01 | 1.3569551 |
| 4.685 | 6.16226277 0.36747191 0.22207078 -0.04627152 | 6.72478e-01 | 1.3692124 |
| 6 | 6.05963406 0.38300155 0.22601257 -0.04677555 | 5.74918e-01 | 1.1370988 |
| 7.315 | 6.16002538 0.36780859 0.22215637 -0.04628246 | 6.70255e-01 | 1.3639225 |
| 8.63 | 6.1072094 0.37578526 0.22418209 -0.0465414 | 6.19035e-01 | 1.2420205 |
| 9.945 | 6.16285209 0.36738213 0.22204804 -0.04626861 | 6.73062e-01 | 1.3706077 |

1,19767.

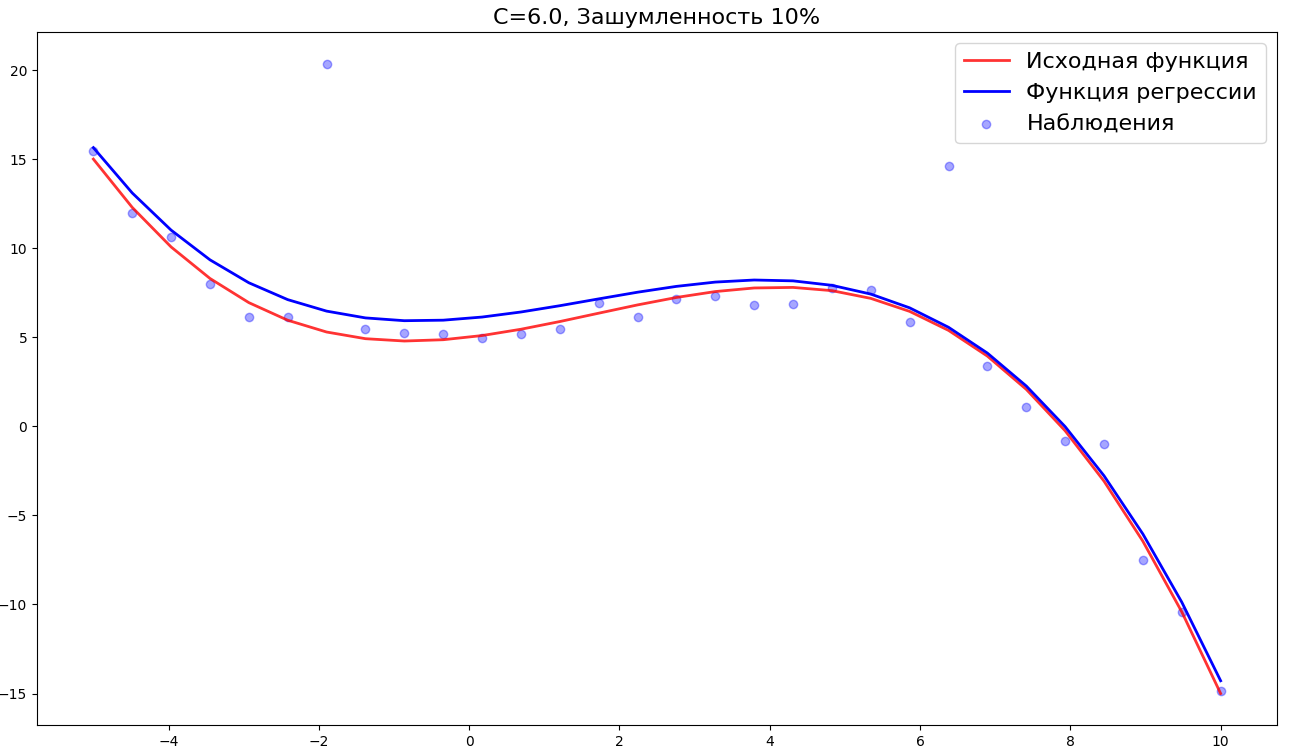


Рисунок 2. Графики наблюдений, функции регрессии и исходной функции для наилучшего значения M-оценки

Засорение 15%.

|  |  |  |  |
| --- | --- | --- | --- |
| **w** | **y** |  |  |
| 1 | 15.458 | 15.726 | 0.2684 |
| 1 | 11.985 | 13.147 | 1.162 |
| 1 | 10.648 | 11.036 | 0.3886 |
| 1 | 7.998 | 9.354 | 1.356 |
| 1 | 6.143 | 8.061 | 1.918 |
| 1 | 6.152 | 7.117 | 0.9649 |
| 0.9697 | 20.365 | 0.6482 | -13.88 |
| 1 | 5.445 | 6.116 | 0.6703 |
| 1 | 5.213 | 5.979 | 0.7661 |
| 1 | 5.204 | 6.032 | 0.8282 |
| 1 | 4.981 | 6.235 | 1.254 |
| 1 | 5.186 | 6.548 | 1.362 |
| 1 | 5.460 | 6.931 | 1.471 |
| 1 | 6.939 | 7.344 | 0.4042 |
| 1 | 6.149 | 7.747 | 1.598 |
| 1 | 7.170 | 8.101 | 0.9306 |
| 1 | 7.314 | 8.365 | 1.051 |
| 1 | 6.791 | 8.500 | 1.709 |
| 1 | 6.872 | 8.467 | 1.595 |
| 1 | 10.923 | 8.224 | -2.699 |
| 1 | 7.647 | 7.732 | 8.526e-02 |
| 1 | 5.865 | 6.952 | 1.088 |
| 0.9951 | 14.628 | 5.844 | -8.784 |
| 1 | 3.382 | 4.367 | 0.9854 |
| 1 | 1.082 | 2.483 | 1.400 |
| 1 | -0.850 | 0.150 | 0.9995 |
| 1 | -0.974 | -2.671 | -1.697 |
| 1 | -7.504 | -6.019 | 1.485 |
| 1 | -10.404 | -9.935 | 0.4694 |
| 1 | -14.849 | -14.458 | 0.3910 |

|  |  |  |  |
| --- | --- | --- | --- |
| С |  | MSE |  |
| 3.37 | 6.1858691 0.43105816 0.22890809 -0.04782688 | 7.63683e-01 | 1.4114881 |
| 4.685 | 6.1532922 0.43611722 0.23018341 -0.04799067 | 7.32758e-01 | 1.3345606 |
| 6 | 6.17807197 0.43227721 0.22921482 -0.04786632 | 7.56224e-01 | 1.3928765 |
| 7.315 | 6.19017955 0.4304028 0.2287419 -0.04780561 | 7.67880e-01 | 1.4218278 |
| 8.63 | 6.1641079 0.43443968 0.22976037 -0.04793635 | 7.42934e-01 | 1.3598593 |
| 9.945 | 6.19028755 0.43038608 0.22873768 -0.04780507 | 7.67984e-01 | 1.4220875 |

-1,46144.

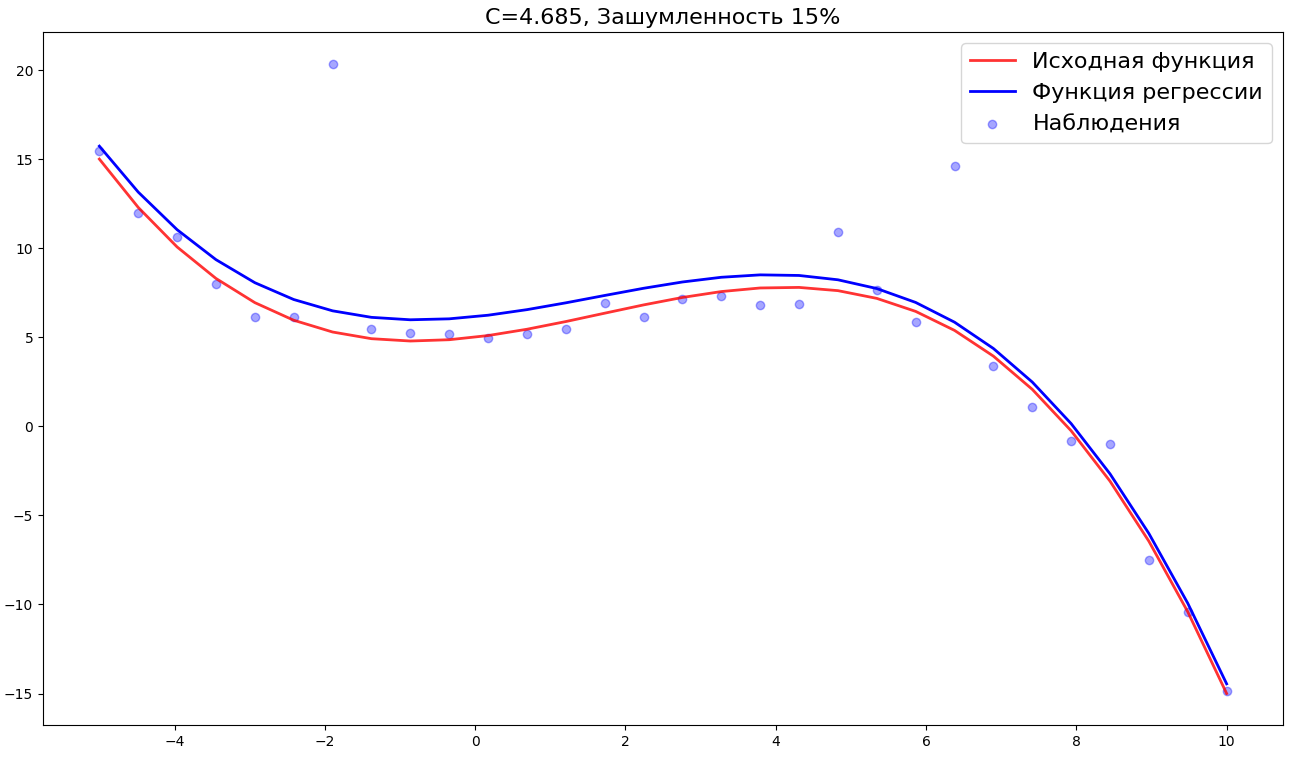


Рисунок 3. Графики наблюдений, функции регрессии и исходной функции для наилучшего значения M-оценки

Засорение 20%.

|  |  |  |  |
| --- | --- | --- | --- |
| **w** | **y** |  |  |
| 1 | 15.458 | 14.740 | -0.7181 |
| 1 | 11.985 | 12.152 | 0.1678 |
| 1 | 10.648 | 10.049 | -0.5987 |
| 0.9961 | 0.781 | 8.388 | 7.607 |
| 1 | 6.143 | 7.128 | 0.9852 |
| 1 | 6.152 | 6.228 | 7.577e-02 |
| 0.9422 | 20.365 | 5.646 | -14.72 |
| 1 | 5.445 | 5.342 | -0.1037 |
| 1 | 5.213 | 5.273 | 6.031e-02 |
| 1 | 5.204 | 5.400 | 0.1959 |
| 0.9998 | 2.284 | 5.680 | 3.396 |
| 1 | 5.186 | 6.073 | 0.8864 |
| 1 | 5.460 | 6.536 | 1.076 |
| 1 | 6.939 | 7.029 | 9.016e-02 |
| 1 | 6.149 | 7.511 | 1.362 |
| 1 | 7.170 | 7.940 | 0.7704 |
| 1 | 7.314 | 8.276 | 0.9613 |
| 1 | 6.791 | 8.475 | 1.684 |
| 1 | 6.872 | 8.499 | 1.627 |
| 0.9999 | 10.923 | 8.305 | -2.618 |
| 1 | 7.647 | 7.851 | 0.2042 |
| 1 | 5.865 | 7.098 | 1.233 |
| 0.9934 | 14.628 | 6.003 | -8.625 |
| 1 | 3.382 | 4.525 | 1.143 |
| 1 | 1.082 | 2.623 | 1.541 |
| 1 | -0.850 | 0.256 | 1.106 |
| 1 | -0.974 | -2.618 | -1.644 |
| 1 | -7.504 | -6.039 | 1.465 |
| 1 | -10.404 | -10.050 | 0.3543 |
| 1 | -14.849 | -14.691 | 0.1583 |

|  |  |  |  |
| --- | --- | --- | --- |
| С |  | MSE |  |
| 3.37 | 5.63757158 0.57648024 0.2328783 -0.04934391 | 3.01825e-01 | 4.1264033e-01 |
| 4.685 | 5.5824306 0.58501617 0.23516752 -0.04963416 | 2.83460e-01 | 3.4667329e-01 |
| 6 | 5.57222487 0.58669078 0.23559014 -0.04968876 | 2.80498e-01 | 3.3516433e-01 |
| 7.315 | 5.62911035 0.57775913 0.23323131 -0.0493883 | 2.98803e-01 | 4.0210787e-01 |
| 8.63 | 5.62872861 0.5778187 0.23324713 -0.04939032 | 2.98670e-01 | 4.0163644e-01 |
| 9.945 | 5.61912872 0.57930988 0.23364489 -0.04944082 | 2.95369e-01 | 3.8987824e-01 |

-0,81482.

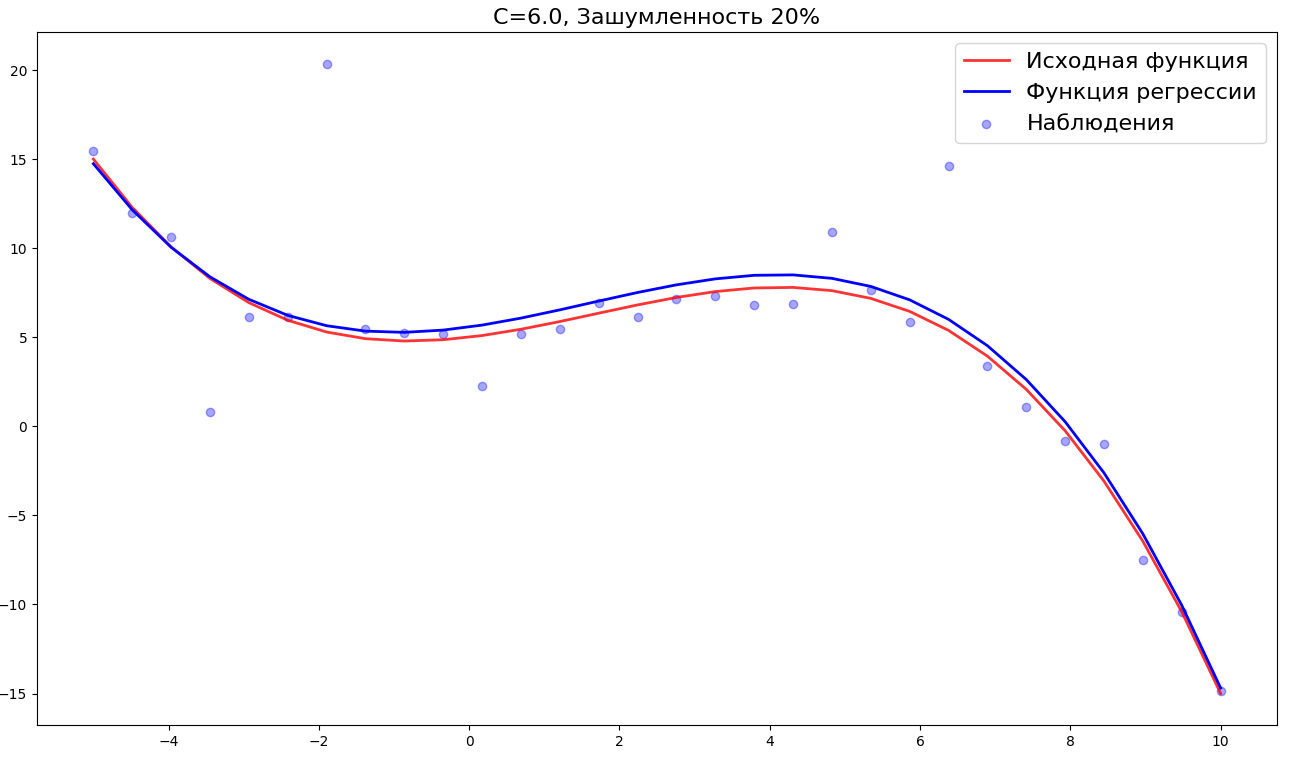


Рисунок 4. Графики наблюдений, функции регрессии и исходной функции для наилучшего значения M-оценки

Засорение 25%.

|  |  |  |  |
| --- | --- | --- | --- |
| **w** | **y** |  |  |
| 1 | 15.458 | 15.227 | -0.2304 |
| 1 | 11.985 | 12.415 | 0.4301 |
| 1 | 10.648 | 10.116 | -0.5315 |
| 1 | 0.781 | 8.288 | 7.507 |
| 1 | 6.143 | 6.889 | 0.7462 |
| 1 | 6.152 | 5.874 | -0.2777 |
| 1 | 20.365 | 5.201 | -15.16 |
| 1 | 5.445 | 4.828 | -0.6174 |
| 1 | 5.213 | 4.711 | -0.5024 |
| 1 | 5.204 | 4.807 | -0.3974 |
| 1 | 2.284 | 5.073 | 2.789 |
| 1 | -1.080 | 5.467 | 6.547 |
| 1 | 5.460 | 5.945 | 0.4849 |
| 1 | 6.939 | 6.464 | -0.4752 |
| 1 | 6.149 | 6.982 | 0.8.329 |
| 1 | 5.677 | 7.456 | 1.778 |
| 1 | 7.314 | 7.842 | 0.5275 |
| 1 | 6.791 | 8.098 | 1.306 |
| 1 | 6.872 | 8.180 | 1.309 |
| 1 | 10.923 | 8.047 | -2.876 |
| 1 | 7.647 | 7.654 | 6.600e-03 |
| 1 | 5.865 | 6.959 | 1.094 |
| 1 | 14.628 | 5.919 | -8.709 |
| 1 | 3.382 | 4.491 | 1.109 |
| 1 | 1.082 | 2.632 | 1.549 |
| 1 | -0.850 | 0.299 | 1.149 |
| 1 | -0.974 | -2.551 | -1.577 |
| 1 | -7.504 | -5.960 | 1.544 |
| 1 | -10.404 | -9.973 | 0.4313 |
| 1 | -14.849 | -14.631 | 0.2181 |

|  |  |  |  |
| --- | --- | --- | --- |
| C |  | MSE |  |
| 3.37 | 4.94973551 0.56797314 0.26551135 -0.05182182 | 1.10028e-01 | 7.3907880e-03 |
| 4.685 | 4.8979711 0.57628443 0.26764878 -0.05209657 | 1.14196e-01 | 1.6545085e-02 |
| 6 | 4.81182098 0.59014967 0.27120664 -0.05255419 | 1.26066e-01 | 4.3994551e-02 |
| 7.315 | 4.96790739 0.5650603 0.26476181 -0.0517255 | 1.09097e-01 | 5.4836664e-03 |
| 8.63 | 4.96617551 0.56533822 0.2648333 -0.05173469 | 1.09175e-01 | 5.6362151e-03 |
| 9.945 | 4.96787138 0.56506608 0.2647633 -0.05172569 | 1.09099e-01 | 5.4867764e-03 |

-0,83090.

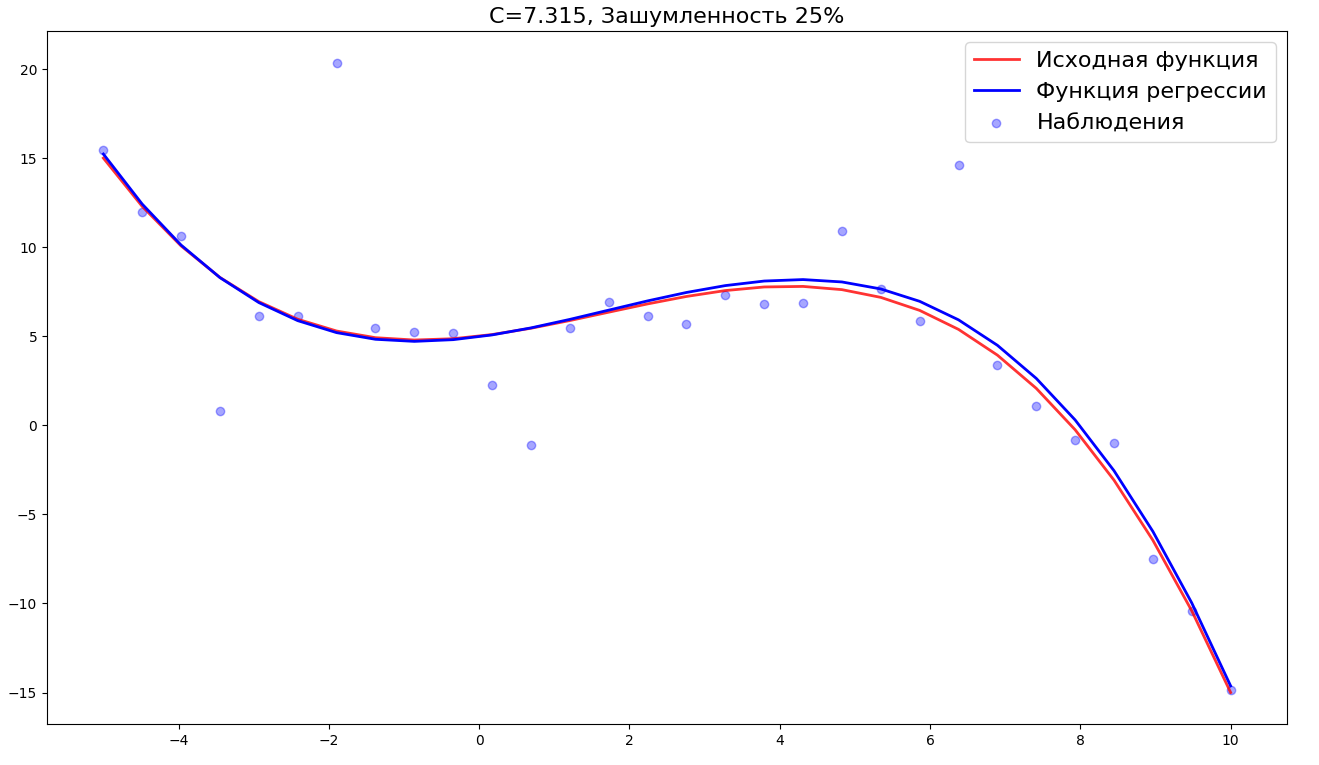
****

Рисунок 5. Графики наблюдений, функции регрессии и исходной функции для наилучшего значения M-оценки

1. **Текст программы:**

**import** argparse  
**import** statistics  
**import** matplotlib.pyplot **as** plt  
**import** pandas **as** pd  
  
**import** numpy **as** np  
**from** tabulate **import** tabulate  
  
**from** my\_types **import** positive\_float, restricted\_int, percent\_float, restricted\_float, required\_length  
  
mad\_const = 0.67449 *# Check - Median absolute deviation*exp\_const = np.exp(1)  
  
params = {**'axes.labelsize'**: 16,  
 **'axes.titlesize'**: 16,  
 **'legend.fontsize'**: 16,  
 **'figure.figsize'**: (16, 9)}  
plt.rcParams.update(params)  
  
  
**def** f(x):  
 **return** np.array([np.ones(len(x)), np.array(x), np.array(x \* x), np.array(x \* x \* x)]).T  
  
  
**def** w(z, c):  
 **if** np.abs(z.all()) < c:  
 **return** 1 - np.power(np.power((z / c), 2), 2)  
 **else**:  
 **return** 0  
  
  
**def** MSE(y, y\_hat):  
 **return** np.square(y\_hat - y).mean()  
  
  
**def** error(theta, theta\_est):  
 diff = theta\_est - theta  
 **return** np.dot(diff.T, diff)  
  
  
**def** generate\_dataset(size, noise\_percent, sigma=1.0, scale=15, verbose=False):  
 lb = -5  
 hb = 10  
 exception\_count = 0  
 theta = np.array([5, 0.5, 0.25, -0.05])  
 x = np.linspace(lb, hb, size)  
 y = np.dot(f(x), theta)  
 y\_hat = y.copy()  
  
 **if** noise\_percent > 0:  
 exception\_count = int(np.round(noise\_percent \* size))  
 gaussian\_noise = np.random.normal(0, sigma, size)  
 **for** pos **in** np.random.choice(size, size=exception\_count, replace=False):  
 gaussian\_noise[pos] \*= scale  
 y\_hat = y\_hat + gaussian\_noise  
  
 **if** verbose:  
 **print**(**"Размер выборки равен: {}"**.format(size))  
 **print**(**"Сгенерировано выбросов: {}"**.format(exception\_count))  
 **return** theta, x, y, y\_hat  
  
  
**def** IRLS(x, y, theta, sigma, c, delta, max\_iter=1000):  
 X = f(x)  
 diagW = []  
 iters = 0  
 **for** i **in** range(max\_iter):  
 r = y - np.dot(X, theta)  
 diagW = w(r / sigma, c)  
 **if**(diagW.all() != 0):  
 W = np.diag(diagW)  
 **else**:  
 W = 0  
  
 theta\_next = np.dot(np.linalg.inv(X.T.dot(W).dot(X)), (X.T.dot(W).dot(y)))  
 sigma = statistics.median(r) / mad\_const  
  
 max\_theta\_diff = np.max(np.abs((theta\_next - theta) / theta))  
 theta = theta\_next  
  
 **if** max\_theta\_diff < delta:  
 iters = i + 1  
 **break  
  
 return** theta, sigma, diagW, iters  
  
  
**def** single\_exec(x, y\_true, y\_noise, c, theta, sigma, delta, noise, plot):  
 theta\_hat, sigma\_hat, diagW, iters = IRLS(x=x, y=y\_noise, theta=theta, sigma=sigma, c=c, delta=delta)  
 y\_hat = np.dot(f(x), theta\_hat)  
  
 mse = MSE(y\_true, y\_hat)  
 theta\_error = error(theta, theta\_hat)  
  
 **if** plot:  
 plt.figure()  
 plt.title(**"С={}, Зашумленность {:.0%}"**.format(c, noise))  
 plt.scatter(x, y\_noise, c=**'b'**, alpha=0.35, label=**"Наблюдения"**)  
 plt.plot(x, y\_true, **'r'**, lw=2, label=**'Исходная функция'**, alpha=0.8)  
 plt.plot(x, y\_hat, **'b'**, lw=2, label=**'Функция регрессии'**)  
 plt.legend()  
 plt.show()  
 **return** theta\_hat, sigma\_hat, mse, theta\_error, diagW, iters  
  
  
**def** main(c\_array, theta\_init, sigma\_init, delta, size, noise\_percent, verbose, plot):  
 table = []  
 hidden\_results = []  
  
 theta\_true, x, y\_true, y\_noise = generate\_dataset(size, noise\_percent, sigma=0.5, verbose=verbose, scale=25)  
  
 **if not** theta\_init:  
 theta\_init = theta\_true  
 **if not** sigma\_init:  
 sigma\_init = statistics.median(y\_noise)  
  
 **if** verbose:  
 *# print(tabulate(y\_noise.reshape(-1, 1), ["Y"], floatfmt='.3f'))* **print**(**'sigma\_init = {}'**.format(sigma\_init))  
 **print**(**'theta\_init = {}'**.format(theta\_init))  
 **print**(**'Среднее = {}'**.format(y\_noise.mean()))  
  
 **for** arg\_c **in** c\_array:  
 theta\_hat, sigma\_hat, mse, theta\_error, diagW, iters = single\_exec(x, y\_true, y\_noise, arg\_c, theta\_init,  
 sigma\_init, delta,  
 noise\_percent,  
 plot)  
 table.append([arg\_c, theta\_hat, mse, theta\_error, iters])  
 hidden\_results.append([diagW, sigma\_hat])  
  
 header = [**"C"**, **"Theta\_Hat"**, **"MSE"**, **"Theta\_Error"**, **"Iters"**]  
 **print**(**"\nOriginal thetas:"**, theta\_true)  
 **print**(tabulate(table, header, tablefmt=**"fancy\_grid"**, floatfmt=(**'g'**, **''**, **'.5e'**, **'.7e'**,)))  
  
 **print**(**'\nЛучшие параметры модели:'**)  
 best\_idx = table.index(min(table, key=**lambda** arg: arg[3]))  
 theta\_hat = table[best\_idx][1]  
  
 **print**(**"C ="**, table[best\_idx][0])  
 **print**(**"Theta\_Hat:"**, theta\_hat)  
 **print**(**"Sigma\_Hat:"**, hidden\_results[best\_idx][1])  
  
 **if** verbose:  
 y\_hat = np.dot(f(x), theta\_hat)  
 data = np.array([hidden\_results[best\_idx][0],  
 y\_noise,  
 y\_hat,  
 (y\_hat - y\_noise)  
 ]).T  
 **print**(tabulate(data, [**"w"**, **"y"**, **"y\_hat"**, **"diff"**], tablefmt=**"fancy\_grid"**,  
 floatfmt=(**'.3e'**, **'.3f'**, **'.3f'**, **'.7e'**,)))  
 df = pd.DataFrame(data=data, columns=[**"w"**, **"y"**, **"y\_hat"**, **"diff"**])  
 df[**'w'**] = df[**'w'**].map(**lambda** param: **'{:.3e}'**.format(param))  
 df[**'y'**] = df[**'y'**].map(**lambda** param: **'{:.3f}'**.format(param))  
 df[**'y\_hat'**] = df[**'y\_hat'**].map(**lambda** param: **'{:.3f}'**.format(param))  
 df[**'diff'**] = df[**'diff'**].map(**lambda** param: **'{:.3e}'**.format(param))  
 df.to\_excel(**"./out.xlsx"**)  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 parser = argparse.ArgumentParser()  
  
 parser.add\_argument(**'-c'**, required=True, type=positive\_float, nargs=**'+'**, help=**'Коэффициент(ы) C.'**)  
 parser.add\_argument(**'--size'**, type=restricted\_int, help=**'Размер выборки. (default: %(default)s)'**, default=64)  
 parser.add\_argument(**'--noise-percent'**, type=percent\_float, help=**'Процент зашумленности (default: %(default)s).'**,  
 default=0.0)  
 parser.add\_argument(**'--theta'**, type=restricted\_float, nargs=**'\*'**,  
 help=**'Начальное приближение theta. (default: Theta Ист.)'**, default=None,  
 action=required\_length(4))  
 parser.add\_argument(**'--sigma'**, type=positive\_float, help=**'Начальная оценка sigma. (default: Медиана Y\_Noise)'**,  
 default=None)  
 parser.add\_argument(**'--delta'**, type=positive\_float,  
 help=**'Условие сходимости для thetas в ИМНК. (default: %(default)s)'**,  
 default=0.05)  
  
 parser.add\_argument(**'-v'**, **'--verbose'**, action=**'store\_true'**, help=**'Включить расширенный лог. (default: %(default)s)'**)  
 parser.add\_argument(**'-p'**, **'--plot'**, action=**'store\_true'**, help=**'Визуализировать результаты. (default: %(default)s)'**)  
 parser.add\_argument(**'--seed'**, type=restricted\_int,  
 help=**'Задает начальные условия для генератора случайных чисел. (default: %(default)s)'**,  
 default=69)  
 args = parser.parse\_args()  
  
 np.random.seed(args.seed)  
  
 main(args.c, args.theta, args.sigma, args.delta, args.size, args.noise\_percent, args.verbose, args.plot)