

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
КАФЕДРА МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ И АНАЛИЗА ДАННЫХ

**Статистическое оценивание параметров линейной
регрессии с выбросами при наличии группирования
наблюдений**

Курсовой проект

Румянцева Андрея Кирилловича
студента 4 курса, специальность
"прикладная математика"

Научный руководитель:
зав. кафедрой ММАД,
канд. физ.-мат. наук, доцент
Бодягин Игорь Александрович

Минск, 2018

Содержание

ВВЕДЕНИЕ	2
1 Модель функции регрессии с аномальными наблюдениями и оценки ее параметров	3
1.1 Матмодель линейной регрессии с выбросами при наличии группирования наблюдений	3
1.2 Метод наименьших квадратов	4
1.3 М-оценки	5
2 Построение оценок параметров регрессии с помощью группирования выборки	6
2.1 Построение функции правдоподобия	7
2.2 Метод секущих	8
2.3 Переклассификация выборки	9
3 Реализация оценок на практике	10
3.1 Моделирование функции регрессии с аномальными наблюдениями	10
3.2 Компьютерные эксперименты с построенными оценками	11
Заключение	15
Список Литературы	16
Приложение	17

ВВЕДЕНИЕ

Существует несколько подходов для оценки параметров регрессии, но далеко не все устойчивы к возникновению аномальных наблюдений, то есть таких наблюдений, которые не подчиняются общей модели. В реальной жизни аномальные наблюдения возникают постоянно. Такие наблюдения могут возникать по разным причинам: из-за ошибки измерения, из-за необычной природы входных данных. По этой причине большинство методов просто неприменимо. В прошлом веке в работах Хьюбера была заложена теория робастного оценивания.

Были предложены следующие робастные оценки[1]:

- М-Оценки
- R-Оценки
- L-Оценки

М-оценки – некоторое подобие оценок максимального правдоподобия (ММП-оценки - частный случай), L-оценки строятся на основе линейных комбинаций порядковых статистик, R-оценки – на основе ранговых статистик.

Будет предложен новый способ оценивания параметров регрессии, где используется группирование выборки, то есть такая модель наблюдений линейной множественной регрессии, когда вместо истинных значений зависимой переменной наблюдаются номера классов (интервалов), в которые попадают эти значения[3]. На практике были полностью реализованы описанные оценки и был произведен анализ оценок.

1 Модель функции регрессии с аномальными наблюдениями и оценки ее параметров

1.1 Матмодель линейной регрессии с выбросами при наличии группирования наблюдений

Введем модель линейной регрессии:

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_n x_{in} + \varepsilon_i, i = \overline{1, N}, \\ y_i &= f(x_i, \beta) + \varepsilon_i, \\ f(x_i, \beta) &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_n x_{in} \end{aligned} \quad (1)$$

или, в векторной форме:

$$y_i = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \dots \\ \beta_n \end{pmatrix} \times \begin{pmatrix} 1 \\ x_{i1} \\ \dots \\ x_{in} \end{pmatrix}^T + \varepsilon_i, \quad (2)$$

где y_i – i -е наблюдение из N наблюдений (N -объем выборки), $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ регрессоры, $\{\beta_k, k = \overline{0, n}\}$ – параметры регрессии, а ε_i – случайная ошибка i -го эксперимента, распределение которой подчиняется нормальному закону с нулевым математическим ожиданием и дисперсией σ^2 .

Пусть вместо истинных значений y_i , заданных формулами (1)-(2), наблюдаются значения с искажениями, описываемыми соотношением:

$$y_i^{\tilde{\varepsilon}} = (\xi_i) y_i + (1 - \xi_i) \eta_i, \quad (3)$$

где ξ_i принимает значение, равное 1, с вероятностью $1 - \tilde{\varepsilon}$ и значение, равное 0, с вероятностью $\tilde{\varepsilon}$, т.е.:

$$\begin{cases} p(\xi_i = 0) = \tilde{\varepsilon}, \\ p(\xi_i = 1) = 1 - \tilde{\varepsilon}, \end{cases}, \quad (4)$$

которая называется функцией линейной регрессии с выбросами. η_i -случайная величина из некоторого вообще говоря неизвестного распределения. Параметр ξ_i имеет следующий содержательный смысл: если $\xi_i = 0$, то вместо истинного значения мы наблюдаем выброс, если $\xi_i = 1$, то наблюдается истинное значение. Переменную $\tilde{\varepsilon}$ будем называть долей аномальных наблюдений. Величины ξ_i, x_i и η_i являются независимыми.

Каждый y_i принадлежит нормальному распределению:

$$y_i = f(x_i, \beta) + \varepsilon_i \sim \mathcal{N}(f(x_i, \beta), \sigma^2). \quad (5)$$

Разделим множество значений функции регрессии, т.е. множество \mathcal{R} , на k полуинтервалов:

$$\mathcal{R} = (-\infty, a_1] \bigcup (a_1, a_2] \bigcup \dots \bigcup (a_{k-1}, +\infty). \quad (6)$$

Обозначим полученные интервалы: ν_0, \dots, ν_{k-1} .

Далее в работе будем считать, что вместо истинных значений зависимых переменных y_i наблюдается только номер класса, к которому это наблюдение попало. Тогда для каждого y_i будем наблюдать лишь номер полуинтервала μ_i , в который он попал.

$$\mu_i = j, \text{ если } y_i \text{ отнесли к полуинтервалу } \nu_j. \quad (7)$$

В данной работе решается задача статистического оценивания параметров модели $\{\beta_k, k = \overline{0, n}\}$ по известным группированным наблюдениям с аномалиями.

1.2 Метод наименьших квадратов

Предположим, что случайные ошибки подчиняются нормальному закону распределения вероятностей:

$$L\{\varepsilon_i\} = N_1(0, \sigma^2), i = \overline{1, n}. \quad (8)$$

Строим логарифмическую функцию правдоподобия. В силу (1) и (2) имеем:

$$L\{y_i\} = N_1(f(x_i; \beta), \sigma^2). \quad (9)$$

Логарифмическая функция правдоподобия выглядит так[2]:

$$l(\beta) = \ln \prod_{i=1}^n \left(\frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(y_i - f(x_i; \beta))^2}{2\sigma^2}} \right) = -\frac{1}{2}n \ln 2\pi\sigma^2 - \frac{1}{2\sigma^2} R^2(\beta), \quad (10)$$

$$R^2(\beta) = \sum_{i=1}^n (\delta y_i)^2 = \sum_{i=1}^n (y_i - f(x_i, \beta))^2 \geq 0. \quad (11)$$

Тогда оценка методом наименьших квадратов из формул (4), (8) такова:

$$\hat{\beta} = \arg \min_{\beta} R^2(\beta). \quad (12)$$

1.3 М-оценки

Швейцарский статистик П.Хьюбер предложил использовать М-оценки [2], которые являются решениями экстремальных задач вида:

$$\sum_{i=1}^n \phi(x_i; \beta) \rightarrow \min_{\beta}, \quad (13)$$

где $\phi(\cdot; \beta)$ -некоторая функция, определяющая конкретный тип оценок и их точность.

Очевидно, что $\phi(\cdot; \beta) \equiv -\ln p(\cdot; \beta)$ дает обычную оценку максимального правдоподобия, построенную по модели без выбросов (1).

Рассмотрим теперь некоторые способы выбора функции $\phi(\cdot; \beta)$ для решения экстремальной задачи в М-оценках.

Для начала определим:

$$u_i = y_i^{\tilde{\varepsilon}} - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_n x_{in}). \quad (14)$$

Тогда существует такие методы[4]:

Способы выбора $\phi(\cdot; \beta)$	
Метод	Целевая функция
Метод Наименьших Квадратов	$\phi(\cdot; \beta)_{OLS} = u^2$
Хьюбера	$\phi(\cdot; \beta)_H = \begin{cases} \frac{1}{2}u^2, u \leq k, \\ k u - \frac{1}{2}k^2, u > k \end{cases}$
Биквадратный	$\phi(\cdot; \beta)_B = \begin{cases} \frac{k^2}{6}(1 - [1 - (\frac{u}{k})^2]^3), u \leq k \\ \frac{k^2}{6}, u > k \end{cases}$

2 Построение оценок параметров регрессии с помощью группирования выборки

Будем работать с моделью регрессии (3), предполагая что имеем регрессию без выбросов (1).

Введем обозначение для функции распределения стандартного нормального закона:

$$\Phi(x) = \frac{1}{\sqrt{2}\sigma} \int_{-\infty}^x e^{\frac{-t^2}{2}} dt. \quad (15)$$

Тогда функцию распределения нормального закона с параметрами μ, σ^2 можно представить как:

$$F(x) = \Phi\left(\frac{x - \mu}{\sigma}\right), \quad (16)$$

где $\sigma = \sqrt{\sigma^2}$.
Обозначим:

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt. \quad (17)$$

Тогда:

$$\Phi(x) = \frac{1}{2} \left[1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right) \right]. \quad (18)$$

Поэтому:

$$F(x) = \frac{1}{2} \left[1 + \text{erf}\left(\frac{x - \mu}{\sqrt{2}\sigma}\right) \right]. \quad (19)$$

При модельных предположениях (5) вероятность попадания y_i в полуинтервал ν_j равна:

$$\begin{aligned} P\{y_i \in \nu_j\} &= F_{y_i}(a_{j+1}) - F_{y_i}(a_j) = \\ &= \begin{cases} \frac{1}{2}(\text{erf}(\frac{a_{j+1}-f(x_i,\beta)}{\sqrt{2}\sigma}) - \text{erf}(\frac{a_j-f(x_i,\beta)}{\sqrt{2}\sigma})), & j = \overline{1, k-2} \\ \frac{1}{2}(1 + \text{erf}(\frac{a_1-f(x_i,\beta)}{\sqrt{2}\sigma})), & j = 0 \\ \frac{1}{2}(1 + \text{erf}(\frac{a_{k-1}-f(x_i,\beta)}{\sqrt{2}\sigma})), & j = k-1 \end{cases}. \end{aligned} \quad (20)$$

Понятно, что:

$$P(\mu_i = j) = P(y_i \in \nu_{\mu_i}). \quad (21)$$

2.1 Построение функции правдоподобия

Составим функцию правдоподобия:

$$l(\beta, \sigma^2, \nu_0, \dots, \nu_{k-1}) = \ln\left(\prod_{i=1}^n P(\mu_i = j)\right) = \quad (22)$$

$$= \sum_{i=1}^n \ln(P(\mu_i = j)). \quad (23)$$

Известно приближение для функции $\text{erf}(x)$:

$$(\text{erf}x)^2 \approx 1 - \exp\left(-x^2 \frac{\frac{4}{\pi} + ax^2}{1 + ax^2}\right), \quad (24)$$

$$a = \frac{8}{3\pi} \frac{3 - \pi}{\pi - 4}.$$

Оно считается достаточно точным для x близких к 0 и к ∞ [7].

Найдем производную для этого приближения:

$$\text{erf}'(x) = \exp\left(-x^2 \frac{\frac{4}{\pi} + ax^2}{1 + ax^2}\right) \frac{-2x \frac{\frac{4}{\pi} + ax^2}{1 + ax^2} + (2ax^3) \frac{\frac{4}{\pi} + ax^2}{1 + ax^2} - \frac{2ax^3}{1 + ax^2}}{2\sqrt{1 - \exp\left(-x^2 \frac{\frac{4}{\pi} + ax^2}{1 + ax^2}\right)}}. \quad (25)$$

Будем максимизировать функцию l . Для этого будем искать нули ее производной. Вычисление будем производить с помощью вычислительных методов (будем использовать метод секущих), так как из-за сложного вида производной вычислить ее аналитически не представляется возможным.

$$\begin{aligned} \frac{\delta l}{\delta \beta} &= \frac{\delta \sum_{i=1}^n \ln(P(\mu_i = j))}{\delta \beta} = \frac{\delta \sum_{i=1}^n \ln P(y_i \in \nu_{\mu_i})}{\delta \beta} = \\ &= \frac{\delta \sum_{i=1}^n \ln\left(\frac{1}{2}\left(\text{erf}\left(\frac{a_{\mu_i+1}-f(x_i, \beta)}{\sqrt{2}\sigma}\right) - \text{erf}\left(\frac{a_{\mu_i}-f(x_i, \beta)}{\sqrt{2}\sigma}\right)\right)\right)}{\delta \beta} = \\ &= \sum_{i=1}^n \left((1 - (\delta_{\mu_i 0} + \delta_{\mu_i k-1})) \frac{(\text{erf}'\left(\frac{a_{\mu_i+1}-f(x_i, \beta)}{\sqrt{2}\sigma}\right) - \text{erf}'\left(\frac{a_{\mu_i}-f(x_i, \beta)}{\sqrt{2}\sigma}\right))}{(\text{erf}\left(\frac{a_{\mu_i+1}-f(x_i, \beta)}{\sqrt{2}\sigma}\right) - \text{erf}\left(\frac{a_{\mu_i}-f(x_i, \beta)}{\sqrt{2}\sigma}\right))} + \right. \\ &\quad \left. + (\delta_{\mu_i 0} + \delta_{\mu_i k-1}) \frac{\text{erf}'\left(\frac{a_{\mu_i}-f(x_i, \beta)}{\sqrt{2}\sigma}\right)}{(1 + \text{erf}\left(\frac{a_{\mu_i}-f(x_i, \beta)}{\sqrt{2}\sigma}\right))} \right) (-1) \frac{\delta f(x_i, \beta)}{\delta \beta} = \quad (26) \end{aligned}$$

$$= - \sum_{i=1}^n \begin{pmatrix} 1 \\ x_{i1} \\ \dots \\ x_{in} \end{pmatrix} \times \left((1 - (\delta_{\mu_i 0} + \delta_{\mu_i k-1})) \frac{(\operatorname{erf}'(\frac{a_{\mu_i+1}-f(x_i, \beta)}{\sqrt{2}\sigma}) - \operatorname{erf}'(\frac{a_{\mu_i}-f(x_i, \beta)}{\sqrt{2}\sigma}))}{(\operatorname{erf}(\frac{a_{\mu_i+1}-f(x_i, \beta)}{\sqrt{2}\sigma}) - \operatorname{erf}(\frac{a_{\mu_i}-f(x_i, \beta)}{\sqrt{2}\sigma}))} + \right. \\ \left. + (\delta_{\mu_i 0} + \delta_{\mu_i k-1}) \frac{\operatorname{erf}'(\frac{a_{\mu_i}-f(x_i, \beta)}{\sqrt{2}\sigma})}{(1 + \operatorname{erf}(\frac{a_{\mu_i}-f(x_i, \beta)}{\sqrt{2}\sigma}))} \right),$$

где δ_{ij} - символ Кронекера.

Доказано, что максимизируя функцию правдоподобия (22), можем получить состоятельную оценку[3] параметров.

Итак, выражение (26) и будем использовать для метода секущих, приближая $\operatorname{erf}'(x)$ с помощью выражения (25).

2.2 Метод секущих

Так как мы не можем привести систему $\frac{\delta l}{\delta \beta} = 0$ к виду, удобному для итерации, то нам придется искать ее нули с помощью метода секущих. Введем вектор ошибки $\tilde{\epsilon}^{(k)} = \beta^* - \beta^{(k)}$. Тогда для его определения имеем:

$$\frac{\delta l(\beta^{(k)} + \tilde{\epsilon}^{(k)})}{\delta \beta} = 0. \quad (27)$$

Строя разложение левой части по формуле Тейлора и ограничиваясь лишь линейными членами[8], будем иметь систему:

$$\frac{\delta}{\delta \beta} \frac{\delta l(\beta^{(k)})}{\delta \beta} \Delta \beta^{(k)} = - \frac{\delta l(\beta^{(k)})}{\delta \beta}. \quad (28)$$

Если матрица $\frac{\delta}{\delta \beta} \frac{\delta l(\beta^{(k)})}{\delta \beta}$ невырожденная (а в нашем случае она диагональная), то из этой системы можно единственным образом найти $\Delta \beta^{(k)}$ и построить приближение:

$$\beta^{(k+1)} = \beta^{(k)} + \Delta \beta^{(k)}. \quad (29)$$

Так как для второй производной l получится довольно сложное выражение, то будем приближать ее с помощью выражения:

$$\frac{\delta}{\delta \beta_j} \frac{\delta l(\beta_1^{(k)}, \dots, \beta_n^{(k)})}{\delta \beta} \approx \frac{\frac{\delta l(\beta_1^{(k)}, \dots, \beta_j^{(k)}, \dots, \beta_n^{(k)})}{\delta \beta} - \frac{\delta l(\beta_1^{(k)}, \dots, \beta_j^{(k-1)}, \dots, \beta_n^{(k)})}{\delta \beta}}{\beta_j^{(k)} - \beta_j^{(k-1)}}. \quad (30)$$

Теперь имеем нули производной функции l , а также ее значения на границе отрезка $[a, b]$. Переберем эти значения и таким образом найдем значение вектора $\hat{\beta}$, где она достигает своего максимального значения.

2.3 Переклассификация выборки

На данном этапе для каждого x_i имеем класс μ_i : т.е. пару (x_i, μ_i) . Теперь попытаемся переклассифицировать выборку. Для этого будем строить новую выборку такого же объема N . Будем идти по каждому элементу (x_i, μ_i) выборки и для этого наблюдения построим новое:

$$(x_i, \check{\mu}_i), \quad (31)$$

где $\check{\mu}_i$ получен по методу L -средних.

$$\check{\mu}_i = \arg \max_j \sum_{k \in V_i, k \neq i} \delta_{\check{\mu}_k j}, \quad (32)$$

где V_i множество индексов l первых L векторов x_l , отсортированных по возрастанию расстояния до вектора x_i .

Итак, переклассифицировав выборку, применим к ней функцию правдоподобия из уравнений (22-23), только используя теперь новые классы $\check{\mu}_i$ вместо μ_i . Аналогично пунктам 2.1-2.2 максимизируем ее и найдем новую оценку параметров $\hat{\beta}$.

3 Реализация оценок на практике

На практике построение оценок является нетривиальной задачей, так как метод секущих имеет свои недостатки. Для метода секущих необходимо, чтобы корни уравнения были отделены, но не существует способа отделения корней в общем случае. Поэтому для решения уравнения нам нужны дополнительные параметры:

- границы отрезка, на котором находится предполагаемая оценка $\hat{\beta}$;
- разность векторов двух начальных приближений $\hat{\beta}^{(1)} - \hat{\beta}^{(0)}$;
- шаг для каждой компоненты $\hat{\beta}_i^{(0)}$.

Тогда решая методом секущих уравнение

$$\frac{\delta l(\beta)}{\delta \beta} = 0 \quad (33)$$

на каждом из отрезков найдем возможные $\hat{\beta}$. Найдя среди полученных приближений $\hat{\beta}$ то, на котором функция правдоподобия (22-23) достигает максимума, найдем решение.

3.1 Моделирование функции регрессии с аномальными наблюдениями

Для начала смоделируем функцию регрессии по методу (3). Для удобства моделируем регрессию с одномерными регрессорами $x_i, i = \overline{1, N}$. Воспользуемся такими параметрами:

Параметры программы	
Переменная	значение
Размер выборки N	1000
Доля выбросов $\tilde{\varepsilon}$	0.1
Параметры регрессии β	$(100, 4)$
Регрессоры x_i	$\sim U(-5, 5)$
ε_i	$\sim N(0, 16)$
η_i	$\sim N(100, 100)$

$U(-5, 5)$ - равномерное распределение на отрезке $[-5, 5]$.

Получаем такой график:

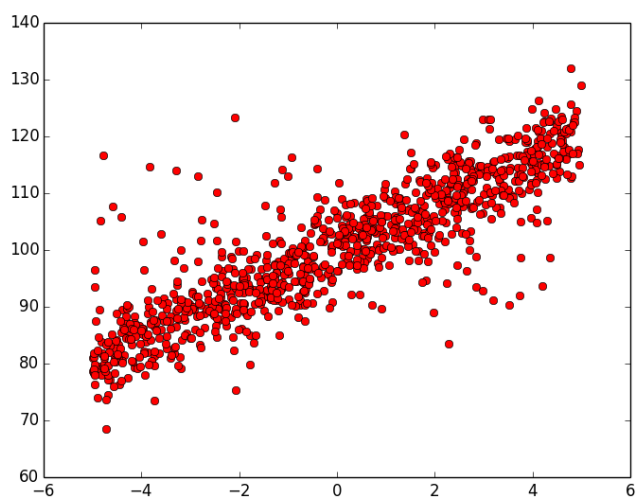


Рис. 1: Вывод графика рассеяния (y_i, x_i)

3.2 Компьютерные эксперименты с построенными оценками

Параметры программы	
Переменная	значение
Размер выборки N	100
Доля выбросов $\tilde{\varepsilon}$	0.8
Параметры регрессии β	$(90, 4)$
Регрессоры x_i	$\sim U(-5, 5)$
ε_i	$\sim N(0, 16)$
η_i	$\sim N(100, 100)$
Величина L из пункта 2.3	10

Построим 100 оценок $\hat{\beta}$, сгенерированных при разных выборках (рис. 2):

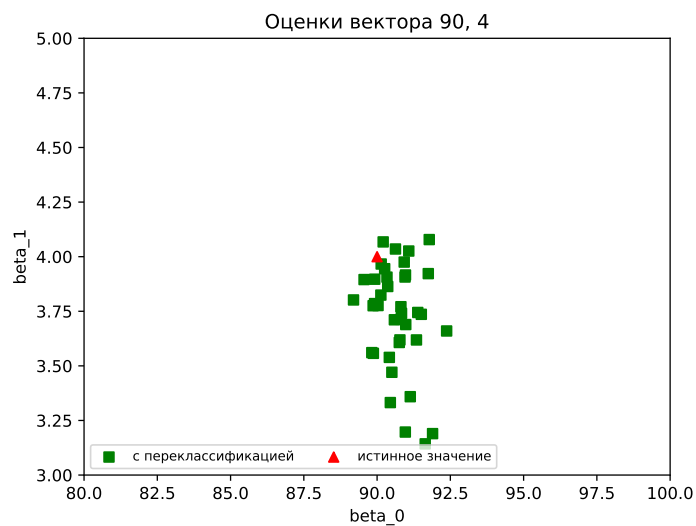


Рис. 2: Вывод графика рассеяния $(\hat{\beta}_0, \hat{\beta}_1)$

Изобразим на рисунке 3 100 оценок $\hat{\beta}$ без переклассификации, описанной в пункте 2.3, и с ней:

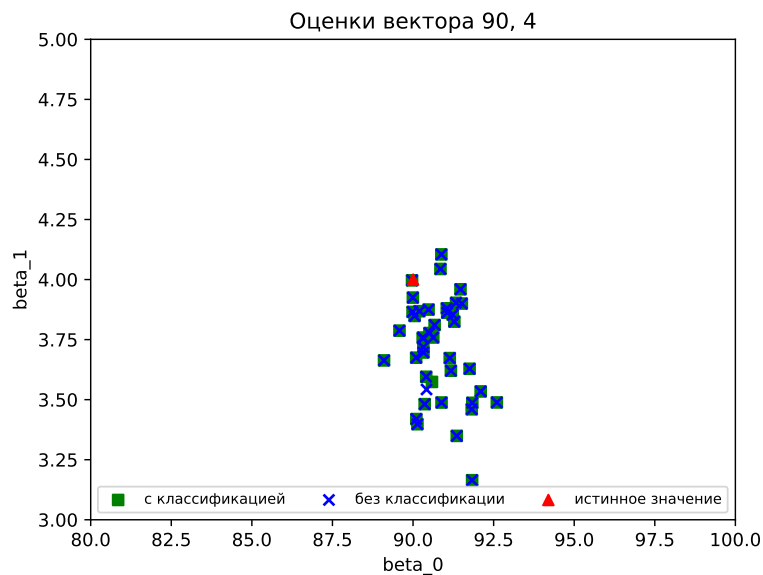


Рис. 3: Вывод графика рассеяния $(\hat{\beta}_0, \hat{\beta}_1)$

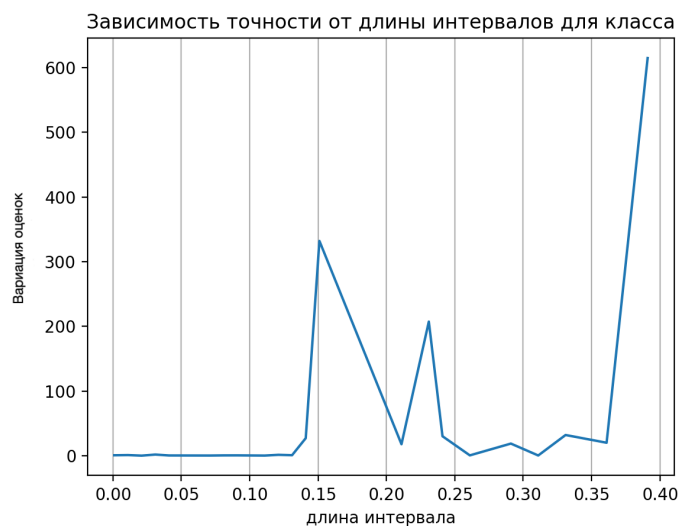


Рис. 4: Зависимость точности от длины интервала

На рисунках 4-5 показана зависимость эмпирической вариации оценок от длины интервалов классификации и объема выборки.



Рис. 5: Зависимость точности от размера выборки

Оценки вектора [90, 4, 7]

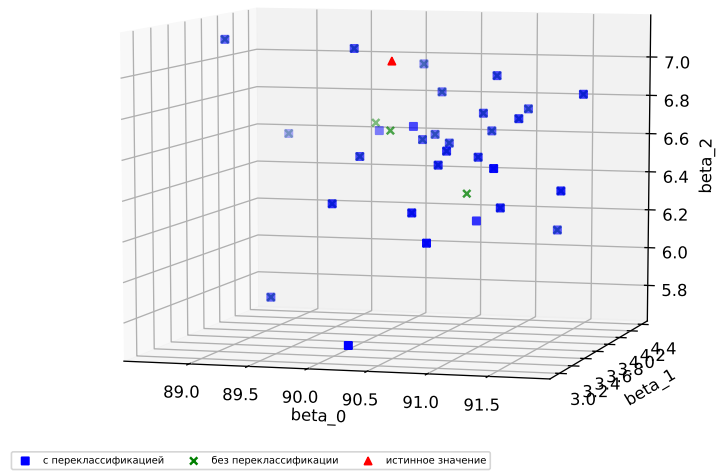


Рис. 6: Вывод графика рассеяния $(\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2)$

Из представленных рисунков можем сделать вывод, что точность увеличивается при увеличении выборки и падает при уменьшении точности классифицирования. Это не опровергает возможную состоятельность оценок.

Заключение

В ходе выполнения курсового проекта были получены следующие результаты:

- рассмотрена математическая модель линейной регрессии с выбросами при наличии группирования наблюдений;
- описаны основные методы оценивания параметров линейной регрессии при наличии выбросов: оценки МНК, М-оценки;
- построены оценки параметров линейной регрессии при наличии группирования наблюдений по методу максимального правдоподобия;
- проведены компьютерные эксперименты в которых построенные оценки применялись к модельным данным;
- результаты экспериментов показали, что построенные оценки могут быть состоятельными.

Список литературы

- [1] Хьюбер Дж П. *Робастность в статистике: пер. с англ.* – М.: Мир, 1984.- 304 с.
- [2] Харин Ю.С., Зуев Н.М., Жук Е.Е. *Теория вероятностей, математическая и прикладная статистика: учебник* – Минск: БГУ, 2011.-463 с.
- [3] Е. С Агеева, чл.-корр. НАН Беларуси Ю.С. Харин *Состоятельность оценки максимального правдоподобия параметров множественной регрессии по классифицированным наблюдениям*
- [4] John Fox, Sanford Weisberg *Robust Regression* – October 8, 2013
- [5] А.В. Омельченко *Робастное оценивание параметров полиномиальной регрессии второго порядка* – Харьковский национальный университет радиоэлектроники, Украина, 2009
- [6] Özlem Gürünlü Alma *Comparison of Robust Regression Methods in Linear Regression* – Int. J. Contemp. Math. Sciences, Vol. 6, 2011, no. 9, 409 - 421 с.
- [7] Sergei Winitzki *A handy approximation for the error function and its inverse.*
- [8] Мандрик П.А., Репников В.И., Фалейчик Б.В., *Численные методы* [Электронный ресурс].

Приложение

Листинг некоторых методов

Метод секущих:

```
def fit_intercept(self, beta_hat=None, beta_hat_next=None):
    if beta_hat is None:
        beta_hat = np.matrix(np.zeros(self.exogen[0].size)).T

    if beta_hat_next is None:
        beta_hat_next = np.matrix(np.ones(self.exogen[0].size)).T

    iteration_counter = 0
    while np.linalg.norm(beta_hat - beta_hat_next) > Defines.METHOD_ACCURACY:
        if self._shared_stop_event.is_set():
            raise StopIteration("fit: got stop event")
        if iteration_counter < Defines.COUNT_LIMIT_OPERATIONS:
            iteration_counter += 1
            dlikelihood_f_for_beta_hat_next = self.full_cl_recl_dlikelihood_f(beta_hat_next)
            delta_beta = np.matrix(np.zeros(self.exogen[0].size)).T

            dlikelihood_derivative_approximation = np.zeros((self.exogen[0].size,
                                                             self.exogen[0].size))

            for i in range(self.exogen[0].size):
                temp_beta = copy.deepcopy(beta_hat_next)
                temp_beta[i] = beta_hat[i]
                dlikelihood_derivative_approximation[i] = ((self.full_cl_recl_dlikelihood_f(
                    beta_hat_next) - self.full_cl_recl_dlikelihood_f(temp_beta)) / (beta_hat_next[i] - beta_hat[i])).A1

            delta_beta = (- np.matrix(dlikelihood_f_for_beta_hat_next)[0] * np.linalg.inv(
                dlikelihood_derivative_approximation))
            beta_hat = beta_hat_next
            beta_hat_next = beta_hat_next + delta_beta.T
        else:
            raise StopIteration("fit: fit_intercept has achieved iteration limit")

    return beta_hat_next
```

Решение методом секущих:

```
def fit(self):
    print("")
    self.classify()
    self.reclassify(Defines.RECLASSIFICATION_LEVEL)

    print("fit: fitting....")

    t_beta_hat = np.matrix([80.0, 0.0]).T
    t_beta_hat_next = np.matrix([100.0, 10.0]).T
    right_bound_indent = Defines.right_bound_fit_indent(self.exogen[0].size)
    loop_indentation_value = Defines.LEFT_BOUND_EVERY_VAR_INDENT
    loop_end_bound = Defines.fit_loop_stop_value(self.exogen[0].size)

    beta_hats_left_bound = Defines.left_bound_fit_init(self.exogen[0].size)

    def recursive_beta_generator(index, previous_step_beta):
        assert (index <= self.exogen[0].size)

        beta_next = np.matrix.copy(previous_step_beta)

        while ((beta_next + right_bound_indent) < loop_end_bound).all():
            if index == self.exogen[0].size:
                yield beta_next
                break

            beta_next[index] += loop_indentation_value

        next_index_generator = recursive_beta_generator(index + 1, beta_next)
        for item in next_index_generator:
```

```

        yield item

fit_intercept_results = []

def fit_intercept_and_add_to_results(beta_hat_one, beta_hat_two):
    t_result = self.fit_intercept(beta_hat_one, beta_hat_two)
    if (np.isnan(t_result) == False).all():
        print("fit: added value to list %s" % t_result)
        fit_intercept_results.append(t_result)
    else:
        raise Exception("fit: got nan")

created_threads = []
for beta_left in recursive_beta_generator(0, beta_hats_left_bound):
    beta_right = beta_left + right_bound_indent
    calculus_thread = Thread(target=fit_intercept_and_add_to_results, args=(np.matrix.copy(beta_left),
                                                                              np.matrix.copy(
                                                                              beta_right)))

    created_threads.append(calculus_thread)
    calculus_thread.start()

for thread in created_threads:
    thread.join(timeout=Defines.THREAD_JOIN_TIMEOUT)

maximum_likelihood_res = None
result_to_return = np.matrix([None for _ in range(self.exogen[0].size)]).T

print("fit: possible betas: ")
print(fit_intercept_results)

initial_left_bound = Defines.left_bound_fit_init(self.exogen[0].size)
initial_right_bound = loop_end_bound

self._shared_stop_event.set()

for result in fit_intercept_results:
    if (result < initial_left_bound).any():
        continue

    if (result > initial_right_bound).any():
        continue

    t_likelihood_res = self.full_cl_recl_likelihood_f(result)
    # print(t_likelihood_res)
    if maximum_likelihood_res is None:
        maximum_likelihood_res = t_likelihood_res
        result_to_return = result

    if maximum_likelihood_res < t_likelihood_res:
        maximum_likelihood_res = t_likelihood_res
        result_to_return = result

return result_to_return

```