МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

КАФЕДРА МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ И АНАЛИЗА ДАННЫХ

Румянцев
Андрей Кирилович

Статистическое оценивание параметров линейной регрессии с выбросами при наличии группирования наблюдений

Дипломная работа

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

Допущена к защите
«__» ____________ 2019 г.
Зав. кафедрой ММАД,
канд. физ.-мат наук, доцент И.А. Бодягин

Минск, 2019

Содержание

РЕФЕРАТ

Дипломная страница, .с, рис., 12 источников.

Ключевые слова: ЛИНЕЙНАЯ РЕГРЕССИЯ, АНОМАЛЬНЫЕ НАБЛЮДЕНИЯ, ГРУППИРОВАНЫЕ НАБЛЮДЕНИЯ, ОЦЕНКИ МАКСИМАЛЬНОГО ПРАВДОПОДОБИЯ.

Объект исследование: линейная регрессия с аномальными наблюдениями при наличии группированых наблюдений. Оценки ее параметров.

Цель работы: предложить способ оценивания параметров линейной регрессии с аномальными наблюдениями при наличии группированых наблюдений, устойчивый к аномальным наблюдениям.

Основные методы исследования: оценки максимального правдоподобия, метод секущих решения систем нелинейных уравнений, наименьший уровень выброса, случайный лес.

ABSTRACT

Graduate work, .pages, figures., 12 sources.

Key words: LINEAR REGRESSION, OUTLIERS, SAMPLE CLUSTERING, MAXIMUM LIKELIHOOD ESTIMATES.

Object of study: linear regression with outliers in the presense of clustered observations . Estimates of its parameters.

Objective: propose a robust method of linear regression with outliers in the presense of clustered observations parameters estimation.

Methods of research: maximum likelihood estimates, secant method of solving nonlinear equations , least outlier factor, random forest.

Result:.

The field of application:.

ВВЕДЕНИЕ

В математической статистике широко используется регрессионная модель. Существует несколько подходов для оценки параметров регрессии, но далеко не все устойчивы к возникновениям 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]:

- M-Оценки
- R-Оценки
- L-Оценки

M-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.

Такие случаи, когда зависимые переменные наблюдаются с выбросами или с пропусками, хорошо исследованы [3]. Более сложный случай, когда вместо содержащих выбросы значений зависимой переменной наблюдаются номера классов(интервалов), в которые попадают эти наблюдения [11].

1 Модель линейной регрессии с выбросами при наличии группирования наблюдений

Рассмотрим модель линейной регрессии:

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

$$y_i = f(x_i, \beta) + \varepsilon_i, \quad (2)$$

$$f(x_i, \beta) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_n x_{in}, \quad (3)$$

Здесь y_i – зависимая переменная, $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ – вектор регрессоров, $\{\beta_k, k = \overline{0, n}\}$– коэффициенты линейной регрессии, а ε_i – случайная ошибка i-го эксперимента, распределение которой подчиняется нормальному закону с нулевым математическим ожиданием и дисперсией σ^2, N-объем выборки. Каждый y_i принадлежит нормальному распределению:

$$y_i = f(x_i, \beta) + \varepsilon_i \sim \mathcal{N}(f(x_i, \beta), \sigma^2). \quad (4)$$

Предполагается, что выборка содержит выбросы, описываемые следующими соотношениями.

$$y_i^{\widetilde{\varepsilon}} = (\xi_i) y_i + (1 - \xi_i) \eta_i, \quad (5)$$

где ξ_i принимает значение, равное 1, с вероятностью $1 - \widetilde{\varepsilon}$ и значение, равное 0, с вероятностью $\widetilde{\varepsilon}$:

$$\begin{cases} P\{\xi_i = 0\} = \widetilde{\varepsilon}, \\ P\{\xi_i = 1\} = 1 - \widetilde{\varepsilon}, \end{cases} , \quad (6)$$

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 $\widetilde{\varepsilon}$ будем называть долей аномальных наблюдений. Величины ξ_i, x_i и η_i являются независимыми.

Пусть множество значений функции регрессии, т.е множество $\mathbb{R}$, разбито на k непересекающихся полуинтервалов:

$$\mathbb{R} = (-\infty, a_1] \bigcup (a_1, a_2] \bigcup \dots \bigcup (a_{k-1}, +\infty). \quad (7)$$

Полученные полуинтервалы будем обозначать: $\nu_0, \ldots, \nu_{k-1}$.

Предполагается, что каждый раз вместо истинного значения зависимой переменной y_i наблюдается только номер интервала, в который это наблюдение попало. Тогда для каждого y_i будем наблюдать лишь номер полуинтервала μ_i, в который он попал.

$$\mu_i = j, \text{если } y_i \in \nu_j. \tag{8}$$

В таком случае принято говорить, что имеет место группирование наблюдений, а сами наблюдения называются группированными [3].

2 Существующие робастные способы оценивания параметров линейной регрессии с выбросами

Для модели регрессии (1) существует несколько робастных способов оценивания параметров. Далее приведены некоторые из них.

2.1 Метод наименьших квадратов

Предлоположим, что случайные ошибки подчиняются нормальному закону распределения вероятностей:

$$L\{\varepsilon_i\} = N_1(0, \sigma^2), i = \overline{1, n}. \tag{9}$$

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

$$L\{y_i\} = N_1(f(x_i; \beta), \sigma^2). \tag{10}$$

Логарифмическая функция правдоподобия выглядит так[2]:

$$l(\beta) = \ln \prod_{i=1}^{n} (\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y_i - f(x_i;\beta))^2}{2\sigma^2}}) \ = \ -\frac{1}{2} n \ln 2\pi\sigma^2 - \frac{1}{2\sigma^2} R^2(\beta), \tag{11}$$

$$R^2(\beta) = \sum_{i=1}^{n} (\delta y_i)^2 \ = \ \sum_{i=1}^{n} (y_i - f(x_i, \beta))^2 \geq 0. \tag{12}$$

Тогда оценка методом наименьших квадратов из формул (6), (9) такова:

$$\hat{\beta} = arg \min_{\beta} R^2(\beta). \tag{13}$$

2.2 М-оценки

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

$$\sum_{i=1}^{n} \phi(x_t; \beta) \to \min_{\beta}, \tag{14}$$

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 (**??**).

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

Для начала определим:

$$u_i = y_i^{\tilde{\varepsilon}} - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_n x_{in}). \tag{15}$$

Тогда существует такие методы[4]:

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

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

Введем обозначение для функции распределения стандартного нормального закона:

$$\Phi(x) = \frac{1}{\sqrt{2}\sigma} \int_{-\infty}^{x} e^{\frac{-t^2}{2}} dt. \tag{16}$$

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

$$F(x) = \Phi(\frac{x-\mu}{\sigma}), \tag{17}$$

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

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-t^2} dt. \tag{18}$$

Тогда:

$$\Phi(x) = \frac{1}{2}\Big[1 + \mathrm{erf}\Big(\frac{x}{\sqrt{2}}\Big)\Big]. \tag{19}$$

Подставив полученные выражения в (17) получим:

$$F(x) = \frac{1}{2}\Big[1 + \mathrm{erf}\Big(\frac{x-\mu}{\sqrt{2}\sigma}\Big)\Big]. \tag{20}$$

При модельных предположениях (4) вероятность попадания y_i в полуинтервал ν_j равна:

$$\begin{aligned} P\{y_i \in \nu_j\} = F_{y_i}(a_{j+1}) - F_{y_i}(a_j) = \\ = \begin{cases} \frac{1}{2}(\mathrm{erf}(\frac{a_{j+1}-f(x_i,\beta)}{\sqrt{2}\sigma}) - \mathrm{erf}(\frac{a_j-f(x_i,\beta)}{\sqrt{2}\sigma})),\ j = \overline{1, k-2} \\ \frac{1}{2}(1 + \mathrm{erf}(\frac{a_1-f(x_i,\beta)}{\sqrt{2}\sigma})),\ j = 0 \\ \frac{1}{2}(1 + \mathrm{erf}(\frac{a_{k-1}-f(x_i,\beta)}{\sqrt{2}\sigma})),\ j = k-1 \end{cases} \end{aligned}. \tag{21}$$

Понятно, что:

$$P(\mu_i = j) = P(y_i \in \nu_{\mu_i}). \tag{22}$$

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

Для этого построим функцию правдоподобия:

$$l(\beta, \sigma^2, \mu_1, \dots, \mu_N) = \sum_{i=1}^{N} \ln(P(y_i \in \nu_{\mu_i})) = \tag{23}$$

$$= \sum_{i=1}^{N} \ln \begin{cases} \frac{1}{2}(\mathrm{erf}(\frac{a_{j+1}-f(x_i,\beta)}{\sqrt{2}\sigma}) - \mathrm{erf}(\frac{a_j - f(x_i,\beta)}{\sqrt{2}\sigma})),\ i = \overline{1, k-2} \\ \frac{1}{2}(1 + \mathrm{erf}(\frac{a_1 - f(x_i,\beta)}{\sqrt{2}\sigma})),\ i = 0 \\ \frac{1}{2}(1 + \mathrm{erf}(\frac{a_{k-1} - f(x_i,\beta)}{\sqrt{2}\sigma})),\ i = k-1 \end{cases}, \tag{24}$$

где:

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt. \tag{25}$$

Для максимизирования функции правдоподобия решим систему уравнений:

$$\frac{\delta l}{\delta \beta} = 0, \tag{26}$$

где:

$$\frac{\delta l}{\delta \beta} = \frac{\delta \sum\limits_{i=1}^{N} \ln P(y_i \in \nu_{\mu_i})}{\delta \beta} =$$

$$= \frac{\delta \sum\limits_{i=1}^{N} \ln(\frac{1}{2}(\mathrm{erf}(\frac{a_{\mu_i+1}-f(x_i,\beta)}{\sqrt{2}\sigma}) - \mathrm{erf}(\frac{a_{\mu_i}-f(x_i,\beta)}{\sqrt{2}\sigma})))}{\delta \beta} =$$

$$= \sum_{i=1}^{N} \Bigg((1 - (\delta_{\mu_i 0} + \delta_{\mu_i k-1})) \frac{(\mathrm{erf'}(\frac{a_{\mu_i+1}-f(x_i,\beta)}{\sqrt{2}\sigma}) - \mathrm{erf'}(\frac{a_{\mu_i}-f(x_i,\beta)}{\sqrt{2}\sigma}))}{(\mathrm{erf}(\frac{a_{\mu_i+1}-f(x_i,\beta)}{\sqrt{2}\sigma}) - \mathrm{erf}(\frac{a_{\mu_i}-f(x_i,\beta)}{\sqrt{2}\sigma}))} +$$
$$+ (\delta_{\mu_i 0} + \delta_{\mu_i k-1}) \frac{\mathrm{erf'}(\frac{a_{\mu_i}-f(x_i,\beta)}{\sqrt{2}\sigma})}{(1 + \mathrm{erf}(\frac{a_{\mu_i}-f(x_i,\beta)}{\sqrt{2}\sigma}))} \Bigg) (-1) \frac{\delta f(x_i, \beta)}{\delta \beta}) = \tag{27}$$

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

δ_{ij} - символ Кронекера.

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

$$\begin{aligned} (\mathrm{erf} x)^2 &\approx 1 - \exp(-x^2 \frac{\frac{4}{\pi} + ax^2}{1 + ax^2}), \\ a &= \frac{8}{3\pi} \frac{3 - \pi}{\pi - 4}. \end{aligned} \quad (28)$$

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

Найдем производную для этого приближения:

$$\mathrm{erf}'(x) = \exp(-x^2 \frac{\frac{4}{\pi} + ax^2}{1 + ax^2}) \frac{-2x \frac{\frac{4}{\pi}+ax^2}{1+ax^2} + (2ax^3) \frac{\frac{4}{\pi}+ax^2}{1+ax^2} - \frac{2ax^3}{1+ax^2}}{2\sqrt{1 - \exp(-x^2 \frac{\frac{4}{\pi}+ax^2}{1+ax^2})}}. \quad (29)$$

Уравнение (26) решается методом секущих.

3.1 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 $\breve{\varepsilon}^{(k)} = \beta^* - \beta^{(k)}$. Тогда для его определения имеем:

$$\frac{\delta l(\beta^{(k)} + \breve{\varepsilon}^{(k)})}{\delta \beta} = 0. \quad (30)$$

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

$$\frac{\delta}{\delta \beta} \frac{\delta l(\beta^{(k)}}{\delta \beta} \Delta \beta^{(k)} = -\frac{\delta l(\beta^{(k)})}{\delta \beta}. \quad (31)$$

Вторая производная функции l приближается с помощью выражения:

$$\frac{\delta}{\delta \beta_j} \frac{\delta l(\beta_1^{(k)}, \ldots, \beta_n^{(k)})}{\delta \beta} \approx \frac{\frac{\delta l(\beta_1^{(k)}, \ldots, \beta_j^{(k)}, \ldots, \beta_n^{(k)})(\beta^{(k)}}{\delta \beta} - \frac{\delta l(\beta_1^{(k)}, \ldots, \beta_j^{(k-1)}, \ldots, \beta_n^{(k)})(\beta^{(k)}}{\delta \beta}}{\beta_j^{(k)} - \beta_j^{(k-1)}}. \quad (32)$$

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:

$$\beta^{(k+1)} = \beta^{(k)} + \Delta \beta^{(k)}. \quad (33)$$

3.2 Переклассификая выборки

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.

Для уменьшения влияния выбросов будем использовать переклассификацию выборки. В ходе выполнения дипломной работы было испробовано несколько способов переклассификации выборки. Один из них: метод K-ближайших соседей.

3.2.1 Метод K-ближайших соседей

Идея заключается в том, что аномальные наблюдения с большей вероятностью попадают не в те интервалы, в которые попадают истинные наблюдения. При этом переклассификация может помочь отнести аномальные наблюдения к истинным классам и улучшить качество оценивания.

На первом этапе для каждого вектора x_i имели класс μ_i: т.е. пару (x_i, μ_i). Далее выполним переклассификацию выборки. Для этого построим новую выборку такого же объема N. Пройдемся по каждому элементу (x_i, μ_i) выборки и для этого наблюдения построим новое:

$$(x_i, \check{\mu}_i), \tag{34}$$

где $\check{\mu}_i$ получен по методу K-соседей.

$$\check{\mu}_i = \arg\max_j \sum_{k \in V_i,\ k \neq i} \delta_{\check{\mu}_k j}\ , \tag{35}$$

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

После переклассификации выборки, к ней применяется функция правдоподобия из уравнений (23), только теперь с использованием новых классов $\check{\mu}_i$ вместо μ_i. Она аналогично максимизируется и в итоге находится новая оценка параметров $\hat{\beta}$.

В ходе экспериментов (раздел 3.2) оказалось, что метод K ближайших соседей показывает слабо исправляет ошибочные классы в выборке, поэтому было решено использовать другой подход в переклассификации.

3.2.2 Переклассификация с использованием Локального уровня выброса и Случайного леса

Идея метода заключается в следующем: определим в выборке такие наблюдения, которые, вероятно, являются аномальными с помощью какого-либо способа. После этого выберем наблюдения, которые не определились как аномальные. Обучим на полученных наблюдениях классификатор. После этого с помощью обученного классификатора переклассифицируем те наблюдения, которые определились как аномальные. В итоге получим выборку, которую будем использовать при решении уравнения (23).

Для определения аномальных наблюдений можно использовать Локальный уровень выброса. Метод был предложен Маркусом М. Бройнигом, Гансом-Петер Кейгелем, Реймондом Т. НГ и Ёргом Сандером в 2000 году. Аномальные наблюдения находятся с помощью измерения локального отклонения точек с учётом их соседей. Метод основан на концепте локальной плотности достижимости, где локальная плотность достижимости вычисляется с учётом ближайших K соседей, расстояние до которых используется для вычисления плотности. Сравнивая локальную плотность точки с локальной плотностью соседей можно определять точки, которые обладают значительно меньшей локальной плотностью по сравнению с соседями. Такие точки будем считать выбросами.

Пусть x_i является некоторой точкой выборки. Определим $\rho(x_i, x_l)$ - расстояние от точки x_i до точки x_l (будем использовать меру Евклида), $\rho_k(x_i)$ - расстояние от точки x_i до её K-го соседа. Аналогично методу K-ближайших соседей обозначим множество индексов k первых K векторов x_k, отсортированных по возрастанию расстояния до вектора x_i через V_i. Использользуя введеные величины зададим:

$$\hat{\rho}_k(x_i, x_l) = \max\{\rho_k(x_l), \rho(x_i, x_l)\} \tag{36}$$

Такую величину можно интерпретировать как истинное расстояние от точки x_l до точки x_i если x_l не входит в ее K ближайших соседей или расстояние до точки $x_{V_{l_K}}$ в противном случае. Данная величина введена для того, чтобы более стабильно вычислять расстояние для тех значений, которые входят в K-соседей точки x_l.

Заметим, что введенная величина не обладает свойством симметричности.

Локальную плотность достижимости зададим выражением:

$$\mathrm{lrd}_k(x_i) = 1/\left(\frac{\sum\limits_{l \in V_i} \hat{\rho}_k(x_i, x_l)}{|V_i|}\right) \tag{37}$$

Теперь можно задать величину:

$$\mathrm{LOF}_k(x_i) = \frac{\sum\limits_{l \in V_i} \frac{\mathrm{lrd}_k(x_l)}{\mathrm{lrd}_k(x_i)}}{|V_i|} \tag{38}$$

Она является средней локальной плотностью достижимости соседей точки x_i по отношению к своей локальной плотности достижимости точки x_i.

По полученному значению можно судить [12], является ли наблюдение x_i выбросом или нет:

- $\mathrm{LOF}_k(x_i) \sim 1$ - означает, что такая же плотность как у соседей

- $\mathrm{LOF}_k(x_i) < 1$ - означает, что большая плотность чем у соседей (наблюдение - истинное)

- $\mathrm{LOF}_k(x_i) > 1$ - означает, что меньшая плотность чем у соседей (наблюдение - выброс)

Найдем аномальные наблюдения в выборке.

Теперь обучим классификатор на предполагаемых истинных значениях выборки. Воспользуемся алгоритмом Random Forest. Построим по выборке несколько деревьев решений. Применим обученный классификатор к каждому аномальному наблюдению. Тот класс, за который проголосовало наибольшее количество простроенных деревьев решений будем считать истинным классом данного аномального наблюдения.

В ходе экспериментов была построена таблица подходящих значений K для заданной длины интервала и объема выборки (раздел 4.7 таблица 4).

3.3 Альтернативные оценки параметров модели

Рассмотрим альтернативный метод оценивания параметров модели регрессии, основанный на замене группированных наблюдений серединами соответствующих интервалов. Такой метод встречается в литературе, например в [9].

Метод заключается в следующем: пусть имеется μ_i - номер полуинтервала, в который попало очередное наблюдение y_i. Ему соответствует полуинтервал ν_{μ_i} (из (8)), т.е. полуинтервал:

$$y_i \in (a_{\nu_{\mu_i}}, a_{\nu_{\mu_i}+1}], i = \overline{1, N} \tag{39}$$

(считаем что $a_1 < y_i < a_{k-1}, i = \overline{1, N}$, т.е $1 \leq \mu_i \leq k-2$).

Найдем центральную точку этого интервала, т.е. точку

$$\check{y}_i = \frac{a_{\nu_{\mu_i}} + a_{\nu_{\mu_i}+1}}{2}. \tag{40}$$

Построим для всех значений функции регрессии y_i значения $\check{y}_i$. Будем использовать в качестве значений функции регрессии полученные значения $\check{y}_i$, а в качестве регрессоров x_i и построим МНК оценки параметров β.

Теперь имеет три вида оценок: оценки максимального правдоподобия, оценки максимального правдоподобия с переклассификацией, МНК по серединам интервалов.

3.4 Полиномиальная регрессия

Введем теперь модель полиномиальной регрессии.

$$\begin{gathered} y_i = \beta_0 + \beta_1 x_{i1}^1 + \beta_2 x_{i2}^2 + \dots + \beta_n x_{in}^n + \varepsilon_i, i = \overline{1, N}, \\ y_i = \sum_{l=1}^{n} x_{il}^{l-1} + \varepsilon_i, i = \overline{1, N}, \\ y_i = f(x_i, \beta) + \varepsilon_i, \\ f(x_i, \beta) = \beta_0 + \beta_1 x_{i1}^1 + \beta_2 x_{i2}^2 + \dots + \beta_n x_{in}^n \end{gathered} \tag{41}$$

В случае полиномиальной регрессии также справедливо:

$$y_i = f(x_i, \beta) + \varepsilon_i \sim \mathcal{N}(f(x_i, \beta), \sigma^2). \tag{42}$$

Поскольку оценки строились путём максимизирования функции:

$$l(\beta, \sigma^2, \nu_0, \dots, \nu_{k-1}) = \textstyle\sum_{i=1}^{N} \ln(P(y_i \in \nu_{\mu_i})) = \tag{43}$$

$$= \sum_{i=1}^{N} \ln \begin{cases} \frac{1}{2}(\mathrm{erf}(\frac{a_{j+1} - f(x_i,\beta)}{\sqrt{2}\sigma}) - \mathrm{erf}(\frac{a_j - f(x_i,\beta)}{\sqrt{2}\sigma})), \ i = \overline{1, k-2} \\ \frac{1}{2}(1 + \mathrm{erf}(\frac{a_1 - f(x_i,\beta)}{\sqrt{2}\sigma})), \ i = 0 \\ \frac{1}{2}(1 + \mathrm{erf}(\frac{a_{k-1} - f(x_i,\beta)}{\sqrt{2}\sigma})), \ i = k-1 \end{cases}, \tag{44}$$

а функция правдоподобия максимизировалась путём решения системы уравнений:

$$\frac{\delta l}{\delta \beta} = 0, \tag{45}$$

которая примет вид:

$$\begin{aligned} \frac{\delta l}{\delta \beta} &= \frac{\delta \sum\limits_{i=1}^{N} \ln P(y_i \in \nu_{\mu_i})}{\delta \beta} = \\ &= \frac{\delta \sum\limits_{i=1}^{N} \ln(\frac{1}{2}(\mathrm{erf}(\frac{a_{\mu_i+1} - f(x_i,\beta)}{\sqrt{2}\sigma}) - \mathrm{erf}(\frac{a_{\mu_i} - f(x_i,\beta)}{\sqrt{2}\sigma})))}{\delta \beta} = \\ &= \sum_{i=1}^{N} \Bigg((1 - (\delta_{\mu_i 0} + \delta_{\mu_i k-1})) \frac{(\mathrm{erf'}(\frac{a_{\mu_i+1} - f(x_i,\beta)}{\sqrt{2}\sigma}) - \mathrm{erf'}(\frac{a_{\mu_i} - f(x_i,\beta)}{\sqrt{2}\sigma}))}{(\mathrm{erf}(\frac{a_{\mu_i+1} - f(x_i,\beta)}{\sqrt{2}\sigma}) - \mathrm{erf}(\frac{a_{\mu_i} - f(x_i,\beta)}{\sqrt{2}\sigma}))} + \\ &\qquad + (\delta_{\mu_i 0} + \delta_{\mu_i k-1}) \frac{\mathrm{erf'}(\frac{a_{\mu_i} - f(x_i,\beta)}{\sqrt{2}\sigma})}{(1 + \mathrm{erf}(\frac{a_{\mu_i} - f(x_i,\beta)}{\sqrt{2}\sigma}))} \Bigg) (-1) \frac{\delta f(x_i, \beta)}{\delta \beta}) = \end{aligned} \tag{46}$$

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

$$+ (\delta_{\mu_i 0} + \delta_{\mu_i k-1}) \frac{\mathrm{erf'}(\frac{a_{\mu_i}-f(x_i,\beta)}{\sqrt{2}\sigma})}{(1 + \mathrm{erf}(\frac{a_{\mu_i}-f(x_i,\beta)}{\sqrt{2}\sigma}))} \Big),$$

то построенные оценки также применимы и для полиномиальной регрессии.

Также как и в случае линейной регрессии считаем, что выборка содержит выбросы, т.е., аналогично:

$$y_i^{\widetilde{\varepsilon}} = (\xi_i) y_i + (1 - \xi_i)\eta_i, \qquad (47)$$

здесь y_i задаются формулой (41).

4 Компьютерные эксперименты

4.1 График рассеяния зависимой пеменной и регрессоров

На следующем изображении представлен график рассеяния зависимой переменной и регрессоров: $(y_i^{\widetilde{\varepsilon}}, x_i)$. Бралась выборка объема $N = 1000$. Доля выбросов $\widetilde{\varepsilon}$ равнялась 0.08. Параметры регрессии выбирались $(90, 4)^T$. Регрессоры x_i выбирались из равномерного распределения $\sim U(-5, 5)$. Ошибки экспериментов подчинялись нормальному закону распределения с параметрами $(0, 16)$. Выбросы подчинялись нормальному распрелению с параметрами $(100, 100)$.

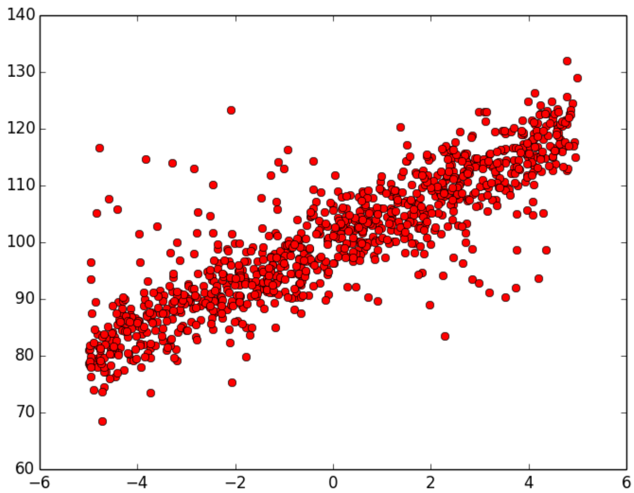

Рис. 1: Вывод графика рассеяния $(y_i^{\widetilde{\varepsilon}}, x_i)$

4.2 Графики рассеяния оценок в случае использования метода K-ближайших соседей для переклассификации

На следующих двух графиках (рис. 2 - рис. 3) изображена диаграмма рассения в случае, когда для переклассификации используется метод K-ближайших соседей.

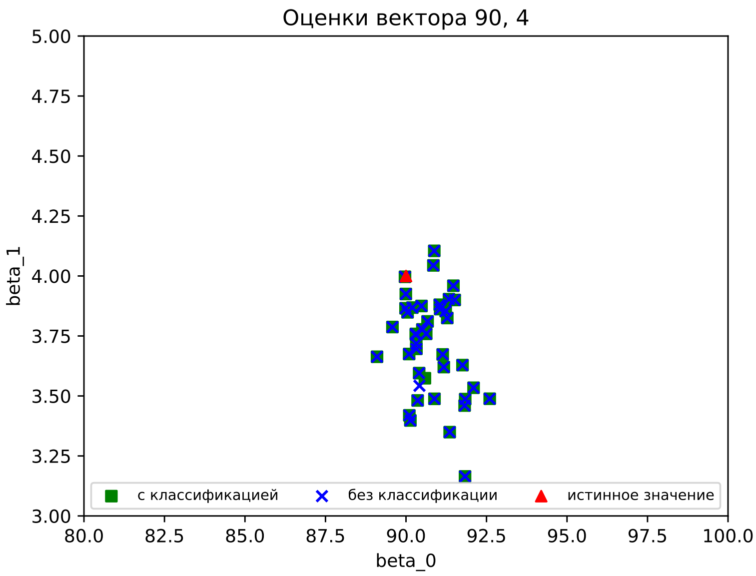

Рис. 2: Вывод графика рассеяния $(\hat{\beta}_0, \hat{\beta}_1)$

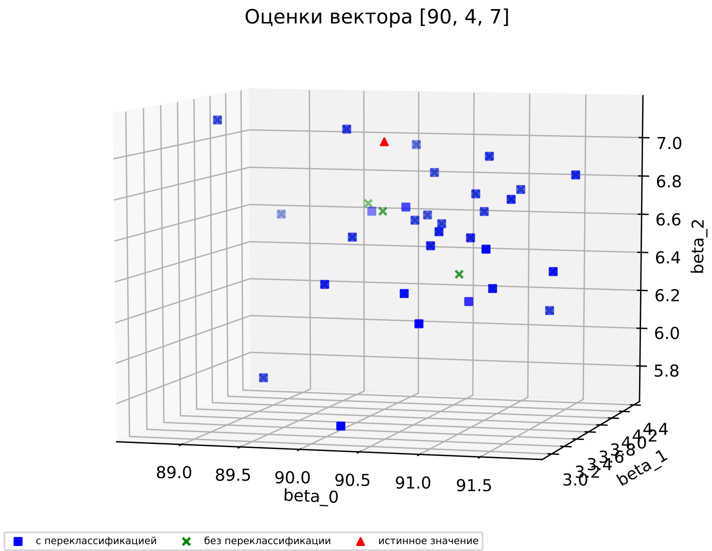

Рис. 3: Вывод графика рассеяния $(\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2)$

4.3 Вариации оценок без переклассификации

В следующем эксперименте переклассификация выборки не использовалась. Объем выборки N изменялся от $N_1 = 140$ до $N_2 = 300$, при этом выборка дополнялась, а не генерировалась новая. Использовалась модель линейной регрессии. Доля выбросов была постоянна и равнялась $\widetilde{\varepsilon} = 0.08$. Регрессоры x_i были из равномерного распределения $U(-5, 5)$. Параметры регрессии были постоянными и равнялись $\beta = (90, 4)^T$. Ошибки эспериментов $\varepsilon_i \sim \mathcal{N}(0, 16)$. На графики изображено изменение вариаций оценок при увеличении выборки.

Рис. 4: Зависимость вариаций от размера выборки

4.4 Сравнение оценок при переклассификации методом K-ближайших соседей с оценками без переклассификации

Были проведены эксперименты для сравнения эмпирической вариации оценок максимального правдоподобия, когда использовалась переклассификация методом K-ближайших соседей и когда не использовалась. При этом на каждой итерации выборка увеличивалась.

Объем выборки N изменялся от $N_1 = 100$ до $N_2 = 400$, при этом выборка дополнялась, а не генерировалась новая. Использовалась модель линейной регрессии. Доля выбросов была постоянна и равнялась $\widetilde{\varepsilon} = 0.08$. Параметры регрессии были постоянными и равнялись $\beta = (90, 4)^T$. Регрессоры x_i были из равномерного распределения $U(-5, 5)$, ошибки эспериментов $\varepsilon_i \sim \mathcal{N}(0, 16)$.

В методе, где использовалась переклассификация, величина K выбиралась: $K = 10$.

Таблица 1: Параметры модели и оценок экспериментов

Параметры программы	
Переменная	значение
Размер выборки N	от 100 до 400
Доля выбросов $\widetilde{\varepsilon}$	0.08
Параметры регрессии β	$(90, 4)$
Регрессоры x_i	$\sim U(-5, 5)$
ε_i	$\sim \mathcal{N}(0, 16)$
η_i	$\sim \mathcal{N}(100, 100)$
В методе, с переклассификацией величина K	10

,

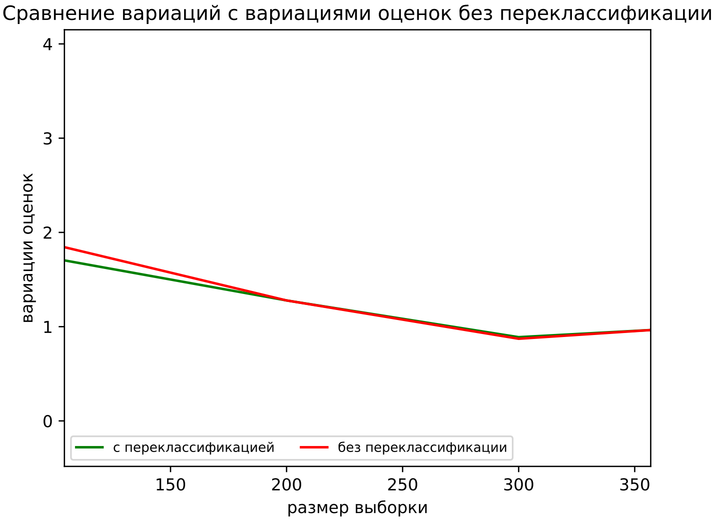

Рис. 5: Сравнение вариаций оценок когда используется и не используется переклассификация

4.5 Сравнительный анализ построенной оценки с альтернативной

4.5.1 Эксперимент с изменением объема выборки

В следующем эксперименте был произведен сравнительный анализ вариаций ОМП-оценок с МНК оценками в зависимости от объема выборки.

Объем выборки N изменялся от $N_1 = 100$ до $N_2 = 500$, при этом выборка дополнялась, а не генерировалась новая. Использовалась модель линейной регрессии. Доля выбросов была постоянна и равнялась $\widetilde{\varepsilon} = 0.08$. Параметры регрессии были постоянными и равнялись $\beta = (90, 4)^T$. Регрессоры x_i были из равномерного распределения $U(-5, 5)$, ошибки эсприментов $\varepsilon_i \sim \mathcal{N}(0, 16)$. Для переклассификации использовался метод K-ближайших соседей.

Таблица 2: Параметры модели и оценок

Параметры программы	
Переменная	значение
Размер выборки N	от 100 до 500
Доля выбросов $\widetilde{\varepsilon}$	0.08
Параметры регрессии β	$(90, 4)$
Регрессоры x_i	$\sim U(-5, 5)$
ε_i	$\sim \mathcal{N}(0, 16)$
η_i	$\sim \mathcal{N}(100, 100)$
Величина K	10

,

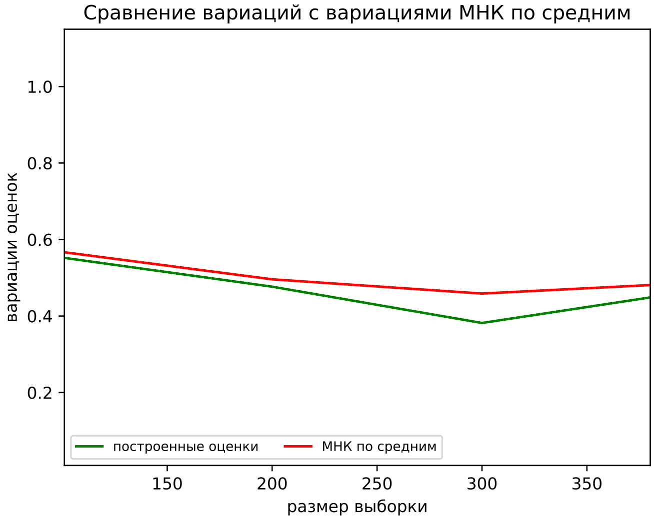

Рис. 6: Сравнение вариаций оценок

При сравнении графиков вариаций (рис.6) можно сделать вывод, что ОМП дают лучший результат,

4.5.2 Эксперимент с полиномиальной регрессией

Был проведен эксперимент с полиномиальной регрессией. Использовались те же параметры модели (таблица 3), объем выборки N изменялся от 100 до 1000:

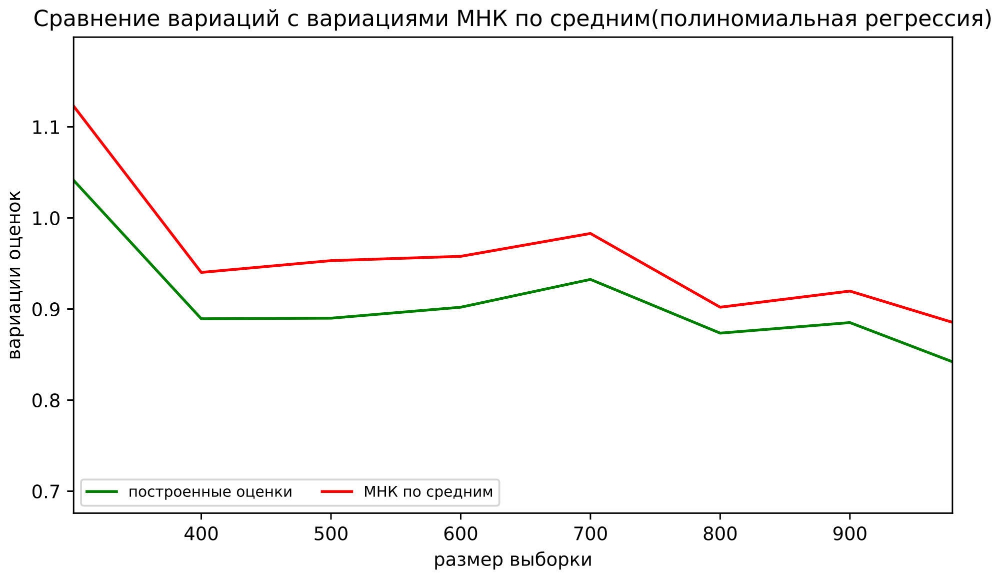

Рис. 7: Вариации оценок в случае полиномиальной регрессии

Оба метода имели схожее поведение при изменении объема выборки, но построенные оценки максимального правдоподобия стабильно показывали лучший результат.

4.6 Эксперименты с изменением уровня переклассификации выборки для метода k-ближайших соседей

В ходе преддипломной практики были построены эксперименты с изменением величины K для метода K-ближайших соседей, используемого в переклассификации.

Объем выборки N был постоянным: $N = 500$. Использовалась модель линейной регрессии. Доля выбросов была постоянна и равнялась $\widetilde{\varepsilon} = 0.08$. Параметры регрессии были постоянными и равнялись $\beta = (90, 4)^T$. Регрессоры x_i были из равномерного распределения $U(-5, 5)$, ошибки эспериментов $\varepsilon_i \sim \mathcal{N}(0, 16)$. Величина K менялась от 10 до 40.

Таблица 3: Параметры модели и оценок экспериментов с переклассификацией выборки

Параметры программы	
Переменная	значение
Размер выборки N	500
Доля выбросов $\widetilde{\varepsilon}$	0.08
Параметры регрессии β	$(90, 4)$
Регрессоры x_i	$\sim U(-5, 5)$
ε_i	$\sim \mathcal{N}(0, 16)$
η_i	$\sim \mathcal{N}(100, 100)$
Величина K	от 10 до 40

,

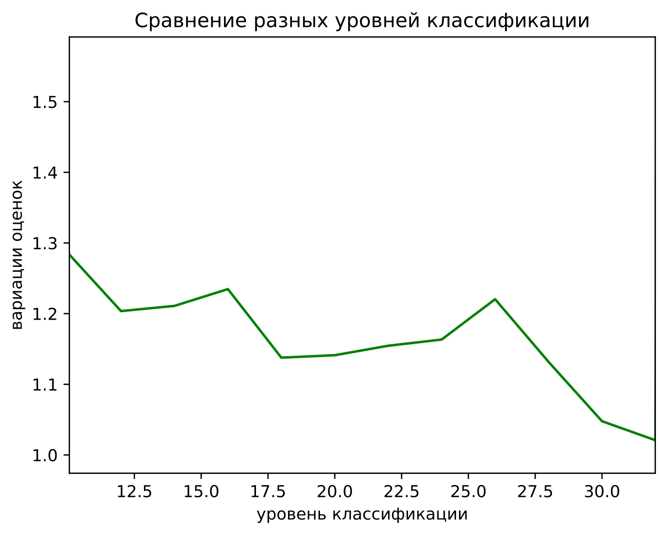

Рис. 8: Зависимость вариаций от K – числа соседей, используемого в переклассификации выборки

В результате получилось, что при увеличении константы K точность оценки параметров растёт.

4.7 Эксперименты с изменением K для Least outlier factor, используемого в переклассификации

В ходе экспериментов была построена таблица, дающая хорошие значения K для выборки определенного объема с определенной длиной интервала, то есть такую величину K, при которой количество неверных классов после переклассификации не более чем такое количество до переклассификации (значения получены опытным путём). Доля выбросов была постоянна и равнялась $\widetilde{\varepsilon} = 0.08$. Параметры регрессии были постоянными и равнялись $\beta = (90, 4)^T$. Регрессоры x_i были из равномерного распределения $U(-5, 5)$, ошибки эспериментов $\varepsilon_i \sim \mathcal{N}(0, 16)$.

Таблица 4: Подходящие значения K для выборки определенного объема с определенной длиной интервала

Объем выборки	длина интервала	величина K
1000	2.0	2
1000	3.0	4
1000	4.0	4
3000	1.0	2
3000	1.5	2
3000	1.75	3
3000	2.0	5
3000	4.0	7
10000	1.0	2
10000	1.5	3
10000	2.0	6
10000	4.0	8

,

Далее приведен пример получаемых в ходе экспериментов значений. было смоделировано 10^5 наблюдений. Процент аномальных наблюдений: 8%. Константа K равнялась 35.

```
modulated with outlier count: 8047
fit: classified
wrong outliers: 7987
fit: reclassificator scored 0.925034 on learning set:
fit: reclassified 10

fit: classified

Classes differ with/without outliers when without reclassification: 7604
Classes differ with/without outliers when with reclassification: 7582
```

Число 8047 - количество выбросов в выборке.

Число 7987 - количество ошибочных выбросов (то есть ошибки рода: отнести выбросы к истинным значениям или истинные значения к выбросам). Хотим, чтобы было по крайней мере меньше предыдущего значения.

Число 7604 - количество неверных значений изначально, то есть неверных значений, полученных извне (это число может быть меньше 8047 так как при достаточно широких классах выбросы могли попасть в нужный класс)

Число 7582 - результат переклассификации, который мы хотим получить меньше предыдущего числа (7604). Чем меньше - тем лучше.

Заключение

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

В ходе дипломной работы был проведен аналитический обзор литературы методов статистического анализа данных при наличии классифицированных наблюдений с искажениями. В результате был реализован альтернативный метод - *метод наименьших квадратов по центрам интервалов.*

Был проведен сравнительный анализ альтернативного метода с оценками максимального правдоподобия. Оценки максимального правдоподобия с переклассификацией выборки показали наилучшие результаты.

Над оценками максимального правдоподобия с переклассификацией выборки были осуществлены эксперименты, в которых изменялась константа K для метода $K-$ соседей (см. п. 4.6). Выяснилось, что увеличение константы K повышает точность аппроксимации.

Реализованные методы максимального правдоподобия с переклассификацией и МНК по серединам интервалов были обобщены на случай полиномиальной регрессии.

По проведенным экспериментам видно, что ОМП с переклассификацией показывают не хуже результаты, чем альтернативные оценки. Можно добиться более точных результатов аппроксимации, если хорошо подобрать параметры оценок.

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 правдопобия 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].

[9] Paolo Giordani *Linear regression analysis for interval-valued data based on the Lasso technique* – Department of Statistical Sciences Sapienza University of Rome

[10] Masahiro Inuiguchi, Tetsuzo Tanino, *interval linear regression methods based on minkowski difference a bridge between traditional and interval linear regression models.* – KYBERNETIKA, volume 42, 2006 , number 4, pages 423 - 440

[11] Nelson, W., Hahn, G.J. *Technometrics.* volume 14, 1972, pages 247–269.

[12] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, Jörg Sander *LOF: Identifying Density-Based Local Outliers* – Proc. ACM SIGMOD 2000 Int. Conf. On Management of Data, pages 93–104, Dalles, TX, 2000

[13] Nathan L. Knight, Jinling Wang *A Comparison of Outlier Detection Procedures and Robust Estimation Methods in GPS Positioning* The University of New South Wales, Journal of Navigation 62(04):699-709, October 2009

Приложение

Метод наименьших квадратов по центрам интервалов

```
class ApproximationGEMModelNaive(ApproximationGEMModelRedesigned):
    def fit(self):
        self.classify()

        def ex_generator(mu_data):
            for i in range(0, self.endogen.size):
                if mu_data[i] is None:
                    continue
                a_mu_i_plus_1 = mu_data[i] * Defines.INTERVAL_LENGTH
                a_mu_i = mu_data[i] * Defines.INTERVAL_LENGTH - Defines.INTERVAL_LENGTH
                yield (a_mu_i_plus_1 + a_mu_i) / 2

        naive_ex_data_positive = np.fromiter(ex_generator(self._np_freq_positive), float)
        naive_ex_data_negative = np.fromiter(ex_generator(self._np_freq_negative), float)

        naive_ex_data_full = np.append(naive_ex_data_positive, naive_ex_data_negative)

        z, resid, rank, sigma = np.linalg.lstsq(self.exogen, naive_ex_data_full, rcond=None)
        return z
```

Моделирование полиномиальной регрессии:

```
def modulate_polynomial_regression(regression_sample_quintity, regression_outlier_percentage):
    regression_parameters = ACCURATE_RESULT
    _x_points = np.zeros(shape=[regression_sample_quintity, len(regression_parameters)])
    _y_points = np.zeros(shape=regression_sample_quintity)

    def np_random_polynomial(size):
        _res = np.zeros(size)
        for i in range(0, size):
            _res[i] = random.uniform(-5, 5) ** (i + 1)

        return _res

    for i in range(0, regression_sample_quintity):
        _x_points[i] = np.append(np.ones(1), np_random_polynomial(len(ACCURATE_RESULT) - 1))
        if random.random() > regression_outlier_percentage / 100:
            _y_points[i] = (_x_points[i] * ACCURATE_RESULT) + np.random.normal(0, 4)
        else:
            _y_points[i] = np.random.normal(100.0, 15.0, size=1)

    return _x_points, _y_points
```

Моделирование линейной регрессии:

```
def modulateRegression(regression_sample_quintity, regression_outlier_percentage):
regression_parameters = ACCURATE_RESULT
_x_points = np.zeros(shape=[regression_sample_quintity, len(regression_parameters)])
_y_points = np.zeros(shape=regression_sample_quintity)

for i in range(0, regression_sample_quintity):
    if random.random() > regression_outlier_percentage / 100:
        _x_points[i] = np.append(np.ones(1), np.random.uniform(-5, 5, size=len(regression_parameters) - 1))
        _y_points[i] = (_x_points[i] * regression_parameters) + np.random.normal(0, 4)
    else:
        _x_points[i] = np.append(np.ones(1), np.random.uniform(-5, 5, size=len(regression_parameters) - 1))
        _y_points[i] = np.random.normal(100.0, 15.0, size=1)

return _x_points, _y_points
```

Метод наименьших квадратов по центрам интервалов:

```
def fit_data_naive_classic():
    sample_sizes = []
    all_results_classic = []
    all_results_naive = []
    for sample_size in range(SAMPLE_SIZE_MIN, SAMPLE_SIZE_MAX+1, SAMPLE_SIZE_STEP):
```

```
        successful_fit = False
        while not successful_fit:
            x_points, y_points = modulateRegression(sample_size, OUTLIER_PERCENTAGE)
            approx_model = groupingEstimates.GEM(x_points, y_points)
            approx_model_naive = groupingEstimatesNaive.GEM_N(x_points, y_points)
            try:
                result = approx_model.fit()
                print("GEM {}".format(result))
                result_naive = approx_model_naive.fit()
                print("GEM_N {}".format(result_naive))

                successful_fit = True

                all_results_classic.append(result)
                all_results_naive.append(result_naive)
                sample_sizes.append(sample_size)
            except KeyboardInterrupt:
                print("stopping...")
                np.save(NP_DATA_PATH + "gem_res_classic", all_results_classic)
                np.save(NP_DATA_PATH + "gem_res_naive", all_results_naive)
                np.save(NP_DATA_PATH + "gem_sizes", sample_sizes)
                quit()
            except Exception as e:
                print(e)
    np.save(NP_DATA_PATH + "gem_res_classic", all_results_classic)
    np.save(NP_DATA_PATH + "gem_res_naive", all_results_naive)
    np.save(NP_DATA_PATH + "gem_sizes", sample_sizes)
```

График с разным объемом выборки:

```
def plot_with_different_sample_size():
    sample_sizes = []
    all_results_with_classification = []
    all_results_without_classification = []

    x_points = None
    y_points = None

    for sample_size in range(SAMPLE_SIZE_MIN, SAMPLE_SIZE_MAX+1, SAMPLE_SIZE_STEP):
        successful_fit = False
        while not successful_fit:
            x_points_t, y_points_t = modulateRegression(sample_size, OUTLIER_PERCENTAGE)

            if x_points is None or y_points is None:
                x_points = x_points_t
                y_points = y_points_t
            else:
                x_points = np.append(x_points, x_points_t, axis=0)
                y_points = np.append(y_points, y_points_t, axis=0)

            approx_model = groupingEstimates.GEM(x_points, y_points)
            try:
                result = approx_model.fit()
                print("GEM {}".format(result))
                result_without = approx_model.fit_without_reclassification()
                print("GEM_without {}".format(result_without))

                successful_fit = True

                all_results_with_classification.append(result)
                all_results_without_classification.append(result_without)
                sample_sizes.append(sample_size)
            except KeyboardInterrupt:
                print("stopping...")
                np.save(NP_DATA_PATH + "gem_res_with", all_results_with_classification)
                np.save(NP_DATA_PATH + "gem_res_without", all_results_without_classification)
                np.save(NP_DATA_PATH + "gem_sizes_with_without", sample_sizes)
                quit()
            except Exception as e:
                print(e)
    np.save(NP_DATA_PATH + "gem_res_with", all_results_with_classification)
    np.save(NP_DATA_PATH + "gem_res_without", all_results_without_classification)
    np.save(NP_DATA_PATH + "gem_sizes_with_without", sample_sizes)
```

График с разным уровнем переклассификации:

```
def plot_with_different_reclassification_level():
    reclassification_levels = []
    all_results_with_classification = []
    recl_level_min = 10
    recl_level_max = 40

    x_points, y_points = modulateRegression(500, OUTLIER_PERCENTAGE)

    for recl_level in range(recl_level_min, recl_level_max + 1, 2):
        GroupingEstimatesDefines.RECLASSIFICATION_LEVEL = recl_level

        successful_fit = False
        while not successful_fit:
            approx_model = groupingEstimates.GEM(x_points, y_points)
            try:
                result = approx_model.fit()
                print("GEM {}".format(result))

                successful_fit = True

                all_results_with_classification.append(result)
                reclassification_levels.append(recl_level)
            except KeyboardInterrupt:
                print("stopping...")
                np.save(NP_DATA_PATH + "gem_with_dif_level_results", all_results_with_classification)
                np.save(NP_DATA_PATH + "gem_with_dif_level_levels", reclassification_levels)
                quit()
            except Exception as e:
                print(e)
    np.save(NP_DATA_PATH + "gem_with_dif_level_results", all_results_with_classification)
    np.save(NP_DATA_PATH + "gem_with_dif_level_levels", reclassification_levels)
```