МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра математического моделирования и анализа данных

КАРАСИК СЕМЁН БОРИСОВИЧ

ПОСТРОЕНИЕ ОПТИМАЛЬНЫХ ПЛАНОВ ЭКСПЕРИМЕНТОВ ДЛЯ ЛИНЕЙНОЙ МНОЖЕСТВЕННОЙ РЕГРЕССИИ С ГЕТЕРОСКЕДАСТИЧЕСКИМИ НАБЛЮДЕНИЯМИ

Дипломная работа

Руководитель
Кирлица Валерий Петрович
доценты кафедры ММАД,
канд. физ.-мат. наук

Рецензент
Крахотко Валерий Васильевич
доценты кафедры МОУ,
канд. физ.-мат. наук

“Допустить к защите”

«___» _____ 2020 г.

Зав. кафедрой ММАД,
канд. физ.-мат. наук, доцент И.А. Бодягин

Минск, 2020

Белорусский государственный университет

Кафедра математического моделирования и анализа данных

ЗАДАНИЕ НА ДИПЛОМНУЮ РАБОТУ

Студент 4 курса 7 группы Карасик Семён Борисович.

1. Тема: «Построение оптимальных планов экспериментов для линейной множественной регрессии с гетероскедастическими наблюденями».
2. Срок представления дипломной работы к защите: 09.06.2020.
3. Исходные данные к дипломной работе
   3.1. Теория оптимального эксперимента (планирование регрессионных экспериментов). Федоров В.В., монография, Главная редакция физико-математической литературы изд-ва "Наука", 1971.
   3.2. Статьи В.П. Кирлица.
   3.3. Теория вероятностей, математическая и прикладная статистика : учебник / Ю. С. Харин, Н. М. Зуев, Е. Е. Жук. – Минск: БГУ, 2011. – 464 с. – (Классическое университетское издание).
4. Содержание дипломной работы
   4.1. Точные D-оптимальные планы экспериментов для модели линейной множественной регрессии с неравноточными наблюдениями для случая 2 и 3 независимых переменных.
   4.2. Точные D-оптимальные планы экспериментов для модели линейной множественной регрессии с линейным изменением.
   4.3. Численный эксперимент для оценки эффективности D-оптимальных планов.

Руководитель дипломной работы _____ В. П. Кирлица

Задание принял к исполнению _____ С. Б. Карасик

Дата _____

АННОТАЦИЯ

В данной дипломной работе рассмотрены вопросы построения точных D-оптимальных планов экспериментов для линейной множественной регрессии с неравноточными наблюдениями. Были получены точные D-оптимальные планы для моделей с двумя и тремя независимыми переменными. Разработана программа на языке программирования Python для размещения заданного числа наблюдений в точках спектра для точных D-оптимальных планов с линейным изменением. Показана эффективность применения D-оптимальных планов при малом числе наблюдений на примере численного эксперимента.

АНАТАЦЫЯ

У дадзенай дыпломнай працы разгледжаны пытанні пабудовы дакладных D-аптымальных планаў эксперыментаў для лінейнай множнай рэгрэсіі з нераўнадакладнымі назіраннямі. Былі атрыманы дакладныя D-аптымальныя планы для мадэляў з двума і трыма незалежнымі зменнымі. Распрацавана праграма на мове праграмавання Python для размяшчэння зададзенага ліку назіранняў у кропках спектру для дакладных D -аптымальных планаў з лінейным змяненнем. Паказана эфектыўнасць прымянення D-аптымальных планаў пры малым ліку назіранняў на прыкладзе колькаснага эксперыменту.

ANNOTATION

This thesis deals with the construction of exact D-optimal experimental designs for linear multiple regression with heteroscedastic observations. Exact D-optimal plans for models with two and three independent variables were obtained. A program in the Python programming language has been developed for placing a given number of observations at points in the spectrum for exact D-optimal plans with linear variation. The effectiveness of using D-optimal plans for a small number of observations is shown using the example of a numerical experiment.

РЕФЕРАТ

Дипломная работа: 49 страниц, 5 иллюстраций, 0 таблиц, 0 источников, 5 приложений.

ТОЧНЫЕ D-ОПТИМАЛЬНЫЕ ПЛАНЫ ЭКСПЕРИМЕНТОВ, ЛИНЕЙНАЯ МНОЖЕСТВЕННАЯ РЕГРЕССИЯ, НЕРАВНОТОЧНЫЕ НАБЛЮДЕНИЯ.

Объект исследования – неравноточные наблюдения. Цель работы – разработать методы построения D-оптимальных планов экспериментов для неравноточных наблюдений.

Методы исследования – методы теории вероятностей, математической статистики, теория оптимального эксперимента.

Результатами являются построенные D-оптимальные планы для неравноточных наблюдений с 2 и 3 независимыми переменными, с линейным изменением.

Областью применения является планирование научных и производственных экспериментов.

Оглавление

ВВЕДЕНИЕ

Эксперимент является неотъемлемой частью научной и практической деятельности, но притом сложной и затратной. Так, произведя новый товар, необходимо узнать его характеристики (например, прочность), с целью чего производится серия экспериментов. Экспериментатор заинтересован в получении наилучших результатов при наименьших затратах, и важную роль в достижении этой цели играет план эксперимента.

В данной работе рассмотрен вопрос построения оптимального плана эксперимента для случая, когда неизвестная зависимость приближается линейной регрессией, а наблюдения гетероскедастичны (неравноточны). Уточним, что для равноточных наблюдений существует развитая теория, а в данной работе показано обобщение теории на случай неравноточных наблюдений.

ГЛАВА 1
ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ ОПТИМАЛЬНОГО ЭКСПЕРИМЕНТА

1.1 Математическая модель эксперимента

Математической моделью эксперимента является регрессионная модель:

$$y = f(x, \theta) + \varepsilon, \tag{1.1}$$

где y – вектор наблюдаемых значений;
x – вектор факторов или контролируемых параметров;
θ – неизвестные параметры;
f – известная функция;
ε – случайная ошибка с $E\{\varepsilon\} = 0$.
То есть эксперимент рассматривается как некоторая функция f, которая зависит от условий проведения эксперимента – вектора x и неизвестных параметров модели – вектора θ, а результат проведения эксперимента есть вектор y, при этом присутствует случая ошибка измерений ε. Задача экспериментатора – наиболее точно оценить неизвестные параметры θ, проведя при этом минимальное число экспериментов.

Модель регрессии в случае, когда функция f линейна по θ, имеет вид

$$y = f(x, \theta) + \varepsilon = \theta_1 f_1(x) + ... + \theta_m f_m(x) + \varepsilon = \theta' f(x) + \varepsilon. \tag{1.2}$$

где y – наблюдаемое значение;
$(x_1, ..., x_m) = x$ – вектор факторов или контролируемых параметров;
$(\theta_1, ..., \theta_m) = \theta$ – неизвестные параметры;
$(f_1(x), ..., f_m(x)) = f(x)$ – известные функции.
ε – случайная ошибка.

Пусть было проведено n наблюдений

$$y_i = \theta' f(x^{(i)}) + \varepsilon_i, i = \overline{1, n}, n \geq m \tag{1.3}$$

где ε_i – ошибки наблюдений:

$$E\{\varepsilon_i\} = 0$$

$$E\{\varepsilon_i \varepsilon_j\} = 0, i \neq j$$

$$D\{\varepsilon_i\} = \sigma_i^2$$

Пусть $\hat{\theta} = T(x^{(1)}, .., x^{(n)})$ – некоторая оценка параметра θ.

Теорема 1 (о наилучшей линейной оценке неизвестного параметра)

$$\hat{\theta} = M^{-1} Y, \tag{1.4}$$

где

$M = \sum_{i=1}^{n} \frac{1}{\sigma_i^2} f(x^i) f'(x^i), |M| \neq 0$ - *информационная матрица Фишера*;

$$Y = \sum_{i=1}^{n} \frac{1}{\sigma_i^2} y_i f(x^i).$$

И дисперсионная матрица оценок $\hat{\theta}$ равна

$$D\{\hat{\theta}\} = M^{-1}. \tag{1.5}$$

Доказательство сформулированной теоремы приведено в [1]. Оценки такого вида являются лучшими в классе несмещенных линейных оценок, так как обладают наименьшей дисперсией в своём классе.

В случае линейной регрессии функции $f_i(x), i = \overline{1, m}$ имеют вид:

$$f_i(x) = \theta_1 x_1 + \theta_1 x_2 + \cdots + \theta_m x_m. \tag{1.6}$$

Обозначим

$$X = \begin{bmatrix} x_{11} & \dots & x_{1m} \\ \dots & \dots & \dots \\ x_{n1} & \dots & x_{nm} \end{bmatrix} \tag{1.7}$$

– матрица плана эксперимента,

$$\varepsilon = (\varepsilon_1, .., \varepsilon_n,) \in R^n, E\{\varepsilon\} = 0, E\{\varepsilon \varepsilon'\} = I_n \tag{1.8}$$

– вектор независимых ошибок со нулевым математическим ожиданием.

Тогда задача (1.1) в случае линейной регрессии (1.6) в матричной форме

имеет вид:

$$y = X\theta + \varepsilon \tag{1.9}$$

Также без ограничения общности для каждого набора x можем считать, что $x_k \in [-1, 1]$, так как линейным преобразованием отрезок $x_k \in [a_k, b_k]$ можно перевести в $[-1, 1]$:

$$z_k = \frac{x_k - (a_k + b_k)/2}{(b_k - a_k)/2}, k = \overline{1, m}$$

1.2 План эксперимента

Экспериментатор заинтересован в проведении эксперимента с наибольшей точностью и минимальными финансовыми и иными затратами. Формализуем понятие точности и затрат.

Обозначим эксперимент буквой ε, а затраты на его проведение – τ, то есть эксперимент есть $\varepsilon(\tau)$. Введём функцию $R(\varepsilon)$ – функция потерь эксперимента $\varepsilon(\tau)$. Будем считать, что эксперимент ε_1 лучше, предпочтительнее эксперимента ε_2, если $R(\varepsilon_1) < R(\varepsilon_2)$.

Точность оценок вычисляется на основании дисперсионной матрицы оценок $D(\varepsilon)$. Точность задаётся как некоторый функционал Ψ от дисперсионной матрицы оценок: $\Psi[D(\varepsilon)]$.

Тогда $R(\varepsilon)$ можно представить в виде:

$$R(\varepsilon) = \tau + \Psi[D(\varepsilon)].$$

Определение 1 *Совокупность величин*

$$x_1, x_2, \ldots, x_n; \tag{1.10}$$

$$r_1, r_2, \ldots, r_n; \tag{1.11}$$

$$\sum_{i=1}^{n} r_i = N \tag{1.12}$$

называется планом (эксперимента) $\varepsilon(N)$, где $x_1, \ldots x_n$ – точки, в которых проводятся наблюдения, $r_1 \ldots r_n$ – число наблюдений в каждой точке, N – общее число наблюдений. Совокупность точек $x_1, x_2, \ldots, x_n$ называется спектром плана $\varepsilon(N)$.

Определение 2 *Нормированным планом $\varepsilon(N)$ называется совокупность величин*

$$x_1, x_2, \ldots, x_n; \tag{1.13}$$

$$p_1, p_2, \ldots, p_n; \tag{1.14}$$

$$\sum_{i=1}^{n} p_i = 1, p_i = \frac{r_i}{N}. \tag{1.15}$$

В условиях, когда N настолько велико, что функцию потерь можно рассматривать как непрерывную по N, вводится понятие непрерывного нормированного плана.

Определение 3 *Непрерывным нормированным планом называется совокупность величин*

$$x_1, x_2, \ldots, x_n; \tag{1.16}$$

$$\rho_1, \rho_2, \ldots, \rho_n; \tag{1.17}$$

$$\sum_{i=1}^{n} \rho_i = 1, \rho_i \in (0, 1). \tag{1.18}$$

Существуют различные критерии оптимальности плана эксперимента. В данной работе будут рассмотрены D-оптимальные планы.

Определение 4 *План эксперимента называется D-оптимальным, когда модуль определителя дисперсионной матрицы оценок параметров минимален:*

$$|\det D\{\hat{\theta}\}| \to \min_{x \in X}. \tag{1.19}$$

Но определитель обратной матрицы обратно пропорционален определителю исходной матрицы:

$$\det D\{\hat{\theta}\} = \det M^{-1}\{\hat{\theta}\} = \frac{1}{\det M(x)}.$$

Таким образом задача построения D-оптимального плана 1.19 свелась к задаче о максимизации модуля определителя информационной матрицы плана:

$$|detM(x)| \to \max_{x \in X}. \tag{1.20}$$

ГЛАВА 2
ОСНОВНЫЕ РЕЗУЛЬТАТЫ

2.1 Точный D-оптимальный план для модели с двумя независимыми переменными

2.1.1 Теорема о непрерывном D-оптимальном плане

Рассмотрим модель наблюдений с двумя независимыми переменными, тремя параметрами, независимыми и неравноточными ошибками наблюдений:

$$y_i = \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \varepsilon(x^{(i)}), i = \overline{1,n}, n \geq 3 \tag{2.1}$$

$$E\{\varepsilon(x^{(i)})\} = 0, E\{\varepsilon^{(i)}\ \varepsilon^{(j)}\} = 0, i \neq j \tag{2.2}$$

$$D\{\varepsilon(x^{(i)})\} = d(x_{i1}, x_{i2}) > 0, \tag{2.3}$$

$$d(x_1, x_2) \geq \frac{\sigma^2}{3}(1 + x_1^2 + x_2^2) \tag{2.4}$$

$$-1 \leq x_{ij} \leq 1, j = \overline{1,2}, i = \overline{1,n}$$

Для дисперсии наблюдений $d(x_1, x_2)$ предполагается, что:

$$d(x_1, x_2) \geq \frac{\sigma^2}{3}(1 + x_1^2 + x_2^2), \sigma > 0. \tag{2.5}$$

Причем (2.5) обращается в равенство в точках спектра плана:

$$\begin{gathered} x^{(1)} = (1,1); \\ x^{(2)} = (-1,1); \\ x^{(3)} = (-1,-1); \\ x^{(4)} = (1,-1) \end{gathered} \tag{2.6}$$

Введём обозначения:

$$d(x^{(1)}) = d_1, d(x^{(2)}) = d_2, d(x^{(3)}) = d_3, d(x^{(4)}) = d_4. \tag{2.7}$$

Теорема 2 *(о непрерывном D-оптимальном плане в точках спектрах плана) Для модели наблюдений* (2.1)-(2.4) *с некоррелированными ошибками наблюдений, имеющими средние значения ноль и дисперсии* $d(x_1, x_2)$*, следующие планы*

являются непрерывными D-оптимальными. План

$$\varepsilon_1^0 = \left\{ \begin{matrix} x^{(1)}, x^{(2)}, x^{(3)} \\ \frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{3} \end{matrix} \right\} \tag{2.8}$$

с дисперсиями наблюдений

$$d(x_1, x_2) \geq \frac{1}{4}(d_1+d_3+2d_1x_1-2d_3x_2-2d_2x_1x_2+(d_1+d_2)x_1^2+(d_2+d_3)x_2^2). \tag{2.9}$$

План

$$\varepsilon_2^0 = \left\{ \begin{matrix} x^{(2)}, x^{(3)}, x^{(4)} \\ \frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{3} \end{matrix} \right\} \tag{2.10}$$

с дисперсиями наблюдений

$$d(x_1, x_2) \geq \frac{1}{4}(d_2+d_4+2d_4x_1+2d_2x_2+2d_3x_1x_2+(d_3+d_4)x_1^2+(d_2+d_3)x_2^2). \tag{2.11}$$

План

$$\varepsilon_3^0 = \left\{ \begin{matrix} x^{(1)}, x^{(3)}, x^{(4)} \\ \frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{3} \end{matrix} \right\} \tag{2.12}$$

с дисперисиями наблюдений

$$d(x_1, x_2) \geq \frac{1}{4}(d_1+d_3-2d_3x_1+2d_1x_2-2d_4x_1x_2+(d_3+d_4)x_1^2+(d_1+d_4)x_2^2). \tag{2.13}$$

План

$$\varepsilon_4^0 = \left\{ \begin{matrix} x^{(1)}, x^{(2)}, x^{(4)} \\ \frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{3} \end{matrix} \right\} \tag{2.14}$$

с дисперисиями наблюдений

$$d(x_1, x_2) \geq \frac{1}{4}(d_2+d_4-2d_2x_1-2d_4x_2+2d_1x_1x_2+(d_1+d_2)x_1^2+(d_1+d_4)x_2^2). \tag{2.15}$$

(2) описывает непрерывный D-оптимальный план для модели (6)-(8) в точках $x^{(3)}, x^{(4)}, x^{(1)}$.

Доказательство. Опишем вначале процесс построения непрерывного D-оптимального плана ε_1^0. Для оптимального плана ε_1^0, по теореме эквивалентно-

сти Кифера - Вольфовица [1], выполняется неравенство:

$$\frac{1}{d(x_1,x_2)}(1,x_1,x_2)M^{-1}(\varepsilon_1^0)\begin{pmatrix}1\\x_1\\x_2\end{pmatrix}\leq 3, |x_1|\leq 1, |x_2|\leq 1, \tag{2.16}$$

где $d(x_1,x_2)$ - непрерывная функция, определяющая дисперсию ошибки наблюдения в точке (x1, x2), $M(\varepsilon_1^0)$ - информационная матрица плана эксперимента. В точках спектра плана ε_1^0 неравенство (2.16), как необходимое условие, обращается в равенство. Исходя из этих условий, построим класс функций $d(x_1,x_2)$, определяющих поведение дисперсии ошибок наблюдений для плана ε_1^0. Информационная матрица плана ε_1^0 равна:

$$M(\varepsilon_1^0)=\frac{1}{3}\left(\frac{1}{d_1}\begin{pmatrix}1\\1\\1\end{pmatrix}(1,1,1)+\frac{1}{d_2}\begin{pmatrix}1\\-1\\1\end{pmatrix}(1,-1,1)+\frac{1}{d_3}\begin{pmatrix}1\\-1\\-1\end{pmatrix}(1,-1,-1)\right)$$
$$=\frac{1}{3}\begin{pmatrix}a&b&c\\b&a&e\\c&e&a\end{pmatrix},$$

где

$$\begin{aligned}a&=d_1^{-1}+d_2^{-1}+d_3^{-1},\\b&=d_1^{-1}-d_2^{-1}-d_3^{-1},\\c&=d_1^{-1}+d_2^{-1}-d_3^{-1},\\d&=d_1^{-1}-d_2^{-1}+d_3^{-1}.\end{aligned} \tag{2.17}$$

Тогда обратная к матрице $M(\varepsilon_1^0)$ имеет вид:

$$M^{-1}(\varepsilon_1^0)=\frac{3}{a^3+2bce-a(b^2+c^2+e^2)}\begin{pmatrix}a^2-e^2,&ce-ab,&be-ac\\ce-ab,&a^2-c^2,&bc-ae\\be-ac,&bc-ae,&a^2-b^2\end{pmatrix}. \tag{2.18}$$

Разрешая неравенство (2.16) относительно $d(x_1,x_2)$ с учетом (2.18) получим класс функций $d(x_1,x_2)$, определяющих изменение дисперсии наблюдений для плана ε_1^0:

$$d(x_1,x_2)\geq f(x_1,x_2), \tag{2.19}$$

где

$$f(x_1,x_2)=\frac{1}{a^3+2bce-a(b^2+c^2+e^2)}\times$$
$$\times\,[a^2-e^2+2(ce-ab)x_1+2(be-ac)x_2+$$
$$2(bc-ae)x_1x_2+(a^2-c^2)x_1^2+(a^2-b^2)x_2^2]. \quad (2.20)$$

Если теперь в функции $f(x_1,x_2)$ вернуться к исходным обозначениям (2.17), то неравенство (2.19) обратится в неравенство (2.8). Необходимое условие оптимальности плана также выполняется, так как в точках спектра плана $x^{(1)},x^{(2)},x^{(3)}$ неравенство (2.8) обращается в равенство.

Справедливость теоремы для планов $\varepsilon_2^0,\varepsilon_3^0,\varepsilon_4^0$ доказывается аналогично, однако меняются обозначения (2.17).

Для ε_2^0:

$$a=d_2^{-1}+d_3^{-1}+d_4^{-1},$$
$$b=-d_2^{-1}-d_3^{-1}+d_4^{-1},$$
$$c=d_2^{-1}-d_3^{-1}-d_4^{-1},$$
$$d=-d_2^{-1}+d_3^{-1}-d_4^{-1};$$

для ε_3^0:

$$a=d_1^{-1}+d_3^{-1}+d_4^{-1},$$
$$b=d_1^{-1}-d_3^{-1}+d_4^{-1},$$
$$c=d_1^{-1}-d_3^{-1}-d_4^{-1},$$
$$d=d_1^{-1}+d_3^{-1}-d_4^{-1};$$

для ε_4^0:

$$a=d_1^{-1}+d_2^{-1}+d_4^{-1},$$
$$b=d_1^{-1}-d_2^{-1}+d_4^{-1},$$
$$c=d_1^{-1}+d_2^{-1}-d_4^{-1},$$
$$d=d_1^{-1}-d_2^{-1}-d_3^{-1};$$

Теорема доказана.

2.1.2 Пример применения теоремы

Рассмотрим план в точках $x^{(3)}, x^{(4)}, x^{(1)}$.

Пусть $d(x_1, x_2)$ такая, что в указанных точках проходит через поверхность $z = 4 + x_1 + x_2$. Тогда

$$\begin{aligned} d_1 &= d(x_1^{(1)}, x_2^{(1)}) = 4 + 1 + 1 = 6; \\ d_2 &= d(x_1^{(2)}, x_2^{(2)}) = 4 - 1 + 1 = 4; \\ d_3 &= d(x_1^{(3)}, x_2^{(3)}) = 4 - 1 - 1 = 2; \\ d_4 &= d(x_1^{(4)}, x_2^{(4)}) = 4 + 1 - 1 = 4. \end{aligned}$$

Используя доказанную теорему, можно получить, что

$$d(x_1, x_2) \geq 2 - x_1 + 3x_2 - 2x_1x_2 + \frac{3}{2}x_1^2 + \frac{5}{2}x_2^2.$$

Канонический вид поверхности (эллиптический параболоид):

$$x^2(4 - \sqrt{5}) + y^2(4 + \sqrt{5}) = 2z$$

График поверхности:

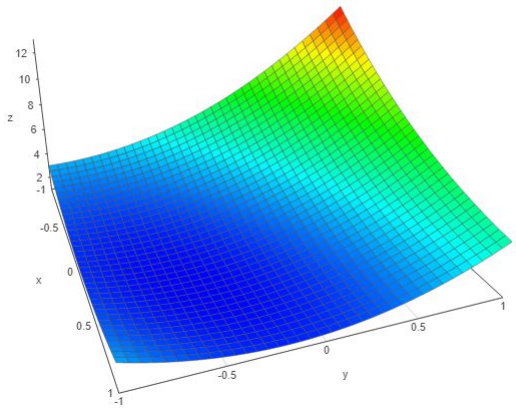

Рис. 2.1: График поверхности отклика первого плана

Используя аналогичные рассуждения, для остальных планов можно получить:

1. Для плана в точках $x^{(1)}, x^{(2)}, x^{(3)}$:

$$d(x_1, x_2) \geq 2 + 3x_1 - x_2 - 2x_1x_2 + \frac{5}{2}x_1^2 + \frac{3}{2}x_2^2.$$

Канонический вид поверхности:

$$x^2(4-\sqrt{5})+y^2(4+\sqrt{5})=2z.$$

График поверхности:

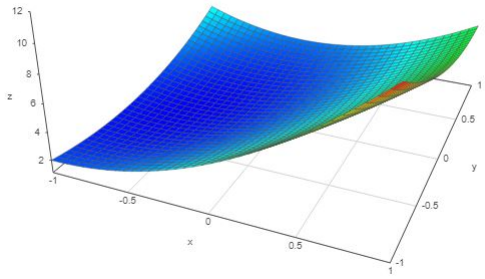

Рис. 2.2: График поверхности отклика второго плана

2. План в точках $x^{(1)}, x^{(2)}, x^{(4)}$:

$$d(x_1,x_2)\geq 2-2x_1-2x_2+3x_1x_2+\frac{5}{2}x_1^2+\frac{5}{2}x_2^2.$$

Канонический вид поверхности:

$$x^2+4y^2=z.$$

График поверхности:

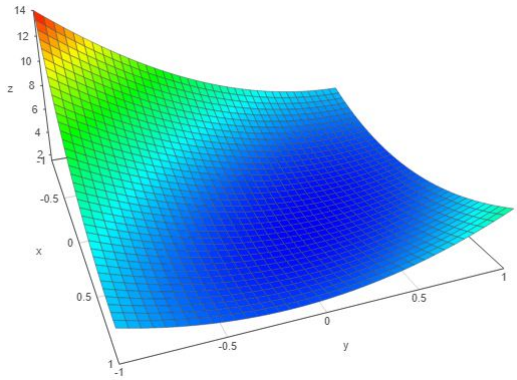

Рис. 2.3: График поверхности отклика третьего плана

3. План в точках $x^{(2)}, x^{(3)}, x^{4)}$:

$$d(x_1,x_2)\geq 2+2x_1+2x_2+x_1x_2+\frac{3}{2}x_1^2+\frac{3}{2}x_2^2.$$

Канонический вид поверхности:

$$x^2 + 2y^2 = z.$$

График поверхности:

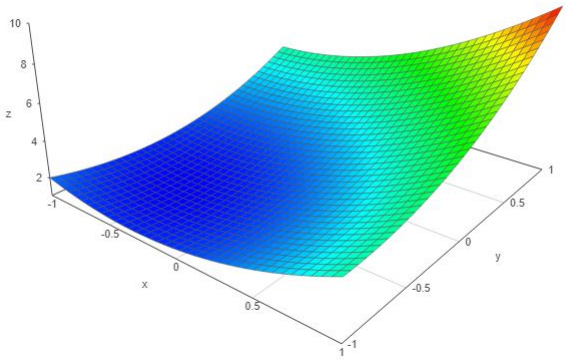

Рис. 2.4: График поверхности отклика четвёртого плана

2.1.3 Программная проверка оптимальности

Подтвердим полученные результаты практически. Для этого была написана программа, которая перебирает все возможные планы исходной задачи и находит оптимальный. Программа написана на языке программирования Python с использованием пакетов *NumPy* [2], *SciPy* [3]. Исходный код программы приведён в приложении А.

Результат работы программы:

```
Best plan and max det(M) for that plan.
Plan 1, 2, 3
[(-1.0, -1.0), (-1.0, 1.0), (1.0, 1.0)] 0.333333333333
Plan 1, 2, 4
[(-1.0, -1.0), (1.0, 1.0), (1.0, -1.0)] 0.333333333333
Plan 1, 3, 4
[(-1.0, -1.0), (1.0, 1.0), (1.0, -1.0)] 0.333333333333
Plan 2, 3, 4
[(-1.0, -1.0), (-1.0, 1.0), (1.0, -1.0)] 0.5
```

Для каждого набора точек программа выводит координаты этих точек и значение определителя при таком наборе точек.

Результат работы программы подтверждает теоретические рассуждения.

2.1.4 Символьная проверка оптимальности

В предыдущем параграфе оптимальность построенных планов была показана численно для заданной функции дисперсии. Покажем символьно, что неравенство (2.16) действительно обращается в равенство в вершинах спектра плана.

Для доказательства этого факта была написана программа в среде Matlab. Указанная программа проверяет, что дисперсия построенных планов в вершинах квадрата в точности обращается в значение дисперсии в заданной точке. Исходный код программы приведён в приложении Б.

Результат работы программы:

```
Plan in points (x2, x3 ,x4)
d(-1, 1)=d2
d(-1, -1)=d3
d(1, -1)=d4

Plan in points (x1, x2 , x3)
d(1, 1)=d1
d(-1, 1)=d2
d(-1, -1)=d3

Plan in points (x1, x2 , x3)
d(1, 1)=d1
d(-1, 1)=d2
d(1, -1)=d4

Plan in points (x1, x3 , x4)
d(1, 1)=d1
d(-1, -1)=d3
d(1, -1)=d4
```

Видно, что равенства выполняются.

2.2 Точный D-оптимальный план для модели с линейным изменением

Рассмотрим следующую модель наблюдений, в которой дисперсия наблюдений имеет линейную зависимость:

$$y_i = \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \varepsilon(x^{(i)}), |x_{ij}| \leq 1, i = \overline{1,n}, j = \overline{1,2} \tag{2.21}$$

$$\begin{aligned} E\{\varepsilon(x^{(i)})\varepsilon(x^{(j)})\} = 0, i \neq j \\ E\{\varepsilon(x^{(i)}) = 0 \end{aligned} \tag{2.22}$$

$$D\{\varepsilon(x^{(i)})\} = a_0 + a_1 x_{i1} + a_2 x_{i2}, a_0 > 0, |a_1| + |a_2| < a_0 \tag{2.23}$$

Теорема 3 . *Для модели наблюдений* (2.21)-(2.23) *существует точный D-оптимальный план, точки спектра которого лежат в вершинах единичного квадрата.*

Доказательство данной теоремы может быть найдено в [4].

Точный D-оптимальный план при заданном n будет иметь следующий вид:

$$\varepsilon^0 = \left\{ \begin{matrix} x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}, \\ n_1 \quad n_2 \quad n_3 \quad n_4 \end{matrix} \right\} \tag{2.24}$$

где n_i - число наблюдений, которое нужно провести в точке $x^{(i)}$.

Однако числа n_1, n_2, n_3, n_4 не известны. Составим программу, которая для заданной функции дисперсии будет находить оптимальное размещение наблюдений в точках спектра плана. Для этого воспользуемся тем фактом, что оптимальный план максимизирует определитель информационной матрицы Фише-

ра $M(\varepsilon)$:

$$M(\varepsilon) = \sum_{i=1}^{4} \frac{n_i}{d_i} \begin{pmatrix} 1 \\ x_{i1} \\ x_{i2} \end{pmatrix} (1, x_{i1}, x_{i2}) =$$

$$= \frac{n_1}{d_1} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} (1,1,1) + \frac{n_2}{d_2} \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} (1,-1,1) +$$

$$+ \frac{n_3}{d_3} \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix} (1,-1,-1) + \frac{n_4)}{d_4} \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} (1,1,-1) =$$

$$= \begin{pmatrix} \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 & \lambda_1 - \lambda_2 - \lambda_3 + \lambda_4 & \lambda_1 + \lambda_2 - \lambda_3 - \lambda_4 \\ \lambda_1 - \lambda_2 - \lambda_3 + \lambda_4 & \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 & \lambda_1 - \lambda_2 + \lambda_3 - \lambda_4 \\ \lambda_1 + \lambda_2 - \lambda_3 - \lambda_4 & \lambda_1 - \lambda_2 + \lambda_3 - \lambda_4 & \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 \end{pmatrix} =$$

$$= \begin{pmatrix} a & b & c \\ b & a & e \\ c & e & a \end{pmatrix},$$

где

$$\lambda_i = \frac{n_i}{d_i},$$

$a, b, c, e -$ соответствующие элемементы матрицы .

Формулы для вычисления определителя подобной матрицы были получены нами ранее.

Тогда для нахождения оптимального плана необходимо перебрать различные наборы n_1, n_2, n_3, n_4 и посмотреть, какие из них максимизируют определитель информационной матрицы.

Описанную логику реализует программа на языке Python, её исходный код может быть найден в приложении В.

Результат работы программы:

```
n = 5

d(x1, x2) >= 40 + 0*x1 + 0*x2
best placements:  [[1, 1, 1, 2], [1, 1, 2, 1], [1, 2,
```

```
   1, 1], [2, 1, 1, 1]]

d(x1, x2) >= 40 + -1*x1 + 0*x2
best placements:  [[1, 1, 1, 2], [1, 1, 2, 1], [1, 2,
   1, 1], [2, 1, 1, 1]]

d(x1, x2) >= 40 + -4*x1 + 0*x2
best placements:  [[1, 1, 1, 2], [1, 1, 2, 1], [1, 2,
   1, 1], [2, 1, 1, 1]]

d(x1, x2) >= 40 + -8*x1 + 0*x2
best placements:  [[1, 1, 1, 2], [2, 1, 1, 1]]

d(x1, x2) >= 40 + -12*x1 + 0*x2
best placements:  [[1, 1, 1, 2], [2, 1, 1, 1]]

d(x1, x2) >= 40 + -30*x1 + 0*x2
best placements:  [[1, 1, 1, 2], [2, 1, 1, 1]]

d(x1, x2) >= 40 + -39*x1 + 0*x2
best placements:  [[1, 1, 1, 2], [2, 1, 1, 1]]

d(x1, x2) >= 40 + -39.5*x1 + 0*x2
best placements:  [[2, 1, 1, 1]]
```

Для каждой поверхности $d(x_1, x_2)$ программа выводит массив оптимальных размещений наблюдений *best placements*, каждый элемент которого есть $[n1, n2, n3, n4]$ - количество наблюдений в точках $x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}$ соответственно.

Проанализируем полученные результаты.

$n = 5$.

В первом случае рассматривается поверхность $d(x_1, x_2) \geq 40 + 0 * x_1 + 0 * x_2$, то есть равноточные наблюдения. В таких условиях существует 4 оптимальных плана, при которых в 3-х вершинах спектра по 1-му наблюдению и 2 наблюдения в 4-ой точке.

Далее рассматривается поверхность $d(x_1, x_2) \geq 40 - 1 * x_1 + 0 * x_2$. Видим, что при незначительном наклоне поверхности оптимальный план не изменяется. Однако в случае $d(x_1, x_2) \geq 40 - 8 * x_1 + 0 * x_2$ существует уже только 2 оптимальных плана: 2 наблюдения размещаются в "менее поднятых" вершинах.
В критическом случае $d(x_1, x_2) \geq 40 - 39.5 * x_1 + 0 * x_2$ существует всего 1 оптимальный план.

2.3 Точный D-оптимальный план для модели с тремя независимыми переменными

В пункте 2.1 была рассмотрена модель линейной регрессии (2.1)-(2.4) с тремя параметрами и двумя независимыми переменными. Обобщим данную модель, вводя третью независимую переменную x_0:

$$y_i = \theta_0 x_{i0} + \theta_1 x_{i1} + \theta_2 x_{i2} + \varepsilon(x^{(i)}), i = \overline{1, n}, n \geq 3 \tag{2.25}$$

$$E\{\varepsilon(x^{(i)})\} = 0, E\{\varepsilon^{(i)}\ \varepsilon^{(j)}\} = 0, i \neq j \tag{2.26}$$

$$D\{\varepsilon(x^{(i)})\} = d(x_{i0}, x_{i1}, x_{i2}) > 0, \tag{2.27}$$

$$d(x_0, x_1, x_2) \geq \frac{\sigma^2}{3}(x_0^2 + x_1^2 + x_2^2), -1 \leq x_{ij} \leq 1, j = \overline{0, 2}, i = \overline{1, n}, \tag{2.28}$$

где n – число наблюдений, x_{ij} – независимые переменные, $x^{(i)} = (x_{i0}, x_{i1}, x_{i2})'$ – вектор наблюдений, $\varepsilon(x^{(i)})$ – случайные некоррелированные ошибки.

Таким образом все наблюдения находятся внутри куба с центром в точке $(0, 0, 0)$ и стороной 2. Занумеруем вершины куба:

$$x^{(1)} = (1, 1, 1), \quad x^{(2)} = (1, -1, 1), \quad x^{(3)} = (1, -1, -1), \quad x^{(4)} = (1, 1, -1),$$
$$x^{(5)} = (-1, 1, 1), \quad x^{(6)} = (-1, -1, 1), \quad x^{(7)} = (-1, -1, -1), \quad x^{(8)} = (-1, 1, -1).$$

Точки $x^{(1)} - x^{(4)}$ лежат на верхней грани куба и обозначены так же, как в модели (2.1)-(2.4).
Также обозначим значение функции дисперсии в вершинах куба:

$$d_i = d(x^{(i)}), i = \overline{1, 8}. \tag{2.29}$$

В теореме 2 было показано, что для модели с двумя независимыми переменными (2.1)-(2.4) можно построить трёхточеный D-оптимальный план. Попро-

буем построить трёхточечный D-оптимальный план для модели с тремя независимыми переменными (2.25)-(2.28).

Теорема 4 . *Для модели неравноточных наблюдений* (2.25)-(2.28) *следующие трехточечные планы являются непрерывными и D-оптимальными.*
План

$$\varepsilon_1^0 = \begin{Bmatrix} x^{(1)}, x^{(2)}, x^{(3)} \\ \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \end{Bmatrix}, \tag{2.30}$$

если для дисперсии наблюдений выполняется:

$$d(x_0, x_1, x_2) \geq \frac{d_2 x_1 x_2}{2} + x_0^2 \left(\frac{d_1}{4} + \frac{d_3}{4} \right) + x_0 \left(\frac{d_1 x_1}{2} - \frac{d_3 x_2}{2} \right) + \\ + x_1^2 \left(\frac{d_1}{4} + \frac{d_2}{4} \right) + x_2^2 \left(\frac{d_2}{4} + \frac{d_3}{4} \right). \tag{2.31}$$

План

$$\varepsilon_2^0 = \begin{Bmatrix} x^{(1)}, x^{(2)}, x^{(4)} \\ \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \end{Bmatrix}, \tag{2.32}$$

если для дисперсии наблюдений выполняется:

$$d(x_0, x_1, x_2) \geq \frac{d_1 x_1 x_2}{2} + x_0^2 \left(\frac{d_2}{4} + \frac{d_4}{4} \right) + x_0 \left(-\frac{d_2 x_1}{2} - \frac{d_4 x_2}{2} \right) + \\ + x_1^2 \left(\frac{d_1}{4} + \frac{d_2}{4} \right) + x_2^2 \left(\frac{d_1}{4} + \frac{d_4}{4} \right). \tag{2.33}$$

План

$$\varepsilon_3^0 = \begin{Bmatrix} x^{(1)}, x^{(3)}, x^{(4)} \\ \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \end{Bmatrix}, \tag{2.34}$$

если для дисперсии наблюдений выполняется:

$$d(x_0, x_1, x_2) \geq \frac{d_4 x_1 x_2}{2} + x_0^2 \left(\frac{d_1}{4} + \frac{d_3}{4} \right) + x_0 \left(\frac{d_1 x_2}{2} - \frac{d_3 x_1}{2} \right) + \\ + x_1^2 \left(\frac{d_3}{4} + \frac{d_4}{4} \right) + x_2^2 \left(\frac{d_1}{4} + \frac{d_4}{4} \right). \tag{2.35}$$

План

$$\varepsilon_4^0 = \begin{Bmatrix} x^{(2)}, x^{(3)}, x^{(4)} \\ \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \end{Bmatrix}, \tag{2.36}$$

если для дисперсии наблюдений выполняется:

$$d(x_0, x_1, x_2) \geq \frac{d_3 x_1 x_2}{2} + x_0^2 \left(\frac{d_2}{4} + \frac{d_4}{4}\right) + x_0 \left(\frac{d_2 x_2}{2} + \frac{d_4 x_1}{2}\right) + \\ + x_1^2 \left(\frac{d_3}{4} + \frac{d_4}{4}\right) + x_2^2 \left(\frac{d_2}{4} + \frac{d_3}{4}\right). \quad (2.37)$$

Доказательство. Доказательство данной теоремы производится аналогично доказательству теоремы 2. Для плана в точках $x^{(i)}, x^{(j)}, x^{(k)}$ запишем критерий эквавалентности Кифера-Вольфовица (2.38):

$$\frac{1}{d(x_0, x_1, x_2)} f'(x) M^{-1}(\varepsilon^0) f(x) \leq 3, \quad (2.38)$$

выразим $d(x_0, x_1, x_2)$ и покажем, что $d(x_0, x_1, x_2)$ равно заданным дисперсиям d_i, d_j, d_k в соответствующих точках.

Ввиду объёмности и однотипности выкладок, описанная схема доказательства была реализована программно при помощи пакета компьютерной алгебры *SymPy* [7], написанном на языке программирования Python. Исходный код программы приведён в приложении Г.

Результат работы программы:

```
plan #1:
d(x0, x1, x2) >= -d2*x1*x2/2 + x0**2*(d1/4 + d3/4) +
   x0*(d1*x1/2 - d3*x2/2) + x1**2*(d1/4 + d2/4) + x2
   **2*(d2/4 + d3/4)
prove that equal in points:
d(x1) = d1
d(x2) = d2
d(x3) = d3

plan #2:
d(x0, x1, x2) >= d1*x1*x2/2 + x0**2*(d2/4 + d4/4) + x0
   *(-d2*x1/2 - d4*x2/2) + x1**2*(d1/4 + d2/4) + x2**2*(
   d1/4 + d4/4)
prove that equal in points:
d(x1) = d1
d(x2) = d2
```

```
d(x4) = d4

plan #3:
d(x0, x1, x2) >= -d4*x1*x2/2 + x0**2*(d1/4 + d3/4) +
   x0*(d1*x2/2 - d3*x1/2) + x1**2*(d3/4 + d4/4) + x2
   **2*(d1/4 + d4/4)
prove that equal in points:
d(x1) = d1
d(x3) = d3
d(x4) = d4

plan #4:
d(x0, x1, x2) >= d3*x1*x2/2 + x0**2*(d2/4 + d4/4) + x0
   *(d2*x2/2 + d4*x1/2) + x1**2*(d3/4 + d4/4) + x2**2*(
   d2/4 + d3/4)
prove that equal in points:
d(x2) = d2
d(x3) = d3
d(x4) = d4
```

Вывод показывает, что неравенства обращаются в равенства в точках спектра плана, что и требовалось доказать. **Теорема доказана**.

Следствие. В теореме 4 были рассмотрены планы, точки которых находятся на верхней грани куба. Куб имеет 6 граней, причём на каждой из них можно построить аналогичные планы. Данный факт следует из того, что точки планов на других гранях эквиваленты точкам одного из планов на верхней грани с точностью до переименования независимых переменных x_0, x_1, x_2 и домножения на -1. Таким образом доказано существование 24-х планов для модели (2.25)-(2.28).

ГЛАВА 3
Численный эксперимент

3.1 Сравнение качества оценок точного D-оптимального и случайного плана

В разделах 2.1 и 2.2 была показана возможность построения точного D-оптимального плана для описанной модели наблюдений. Конечной целью эксперимента является оценка параметров модели по полученным наблюдениям. Далее исследуем: при каких условиях и на сколько оценки, полученные по наблюдениям в условиях D-оптимального плана, точнее оценок, полученных в результате наблюдений в случайных точках?

3.1.1 Модель наблюдений

Рассмотрим модель наблюдений из 2.1. Пусть было проведено N наблюдений. Модель наблюдений имеет вид:

$$y_i = \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \varepsilon(x^{(i)}), i = \overline{1, N}, N \geq 3; \tag{3.1}$$

$$E\{\varepsilon(x^{(i)})\} = 0, E\{\varepsilon^{(i)}\,\varepsilon^{(j)}\} = 0, i \neq j; \tag{3.2}$$

$$D\{\varepsilon(x^{(i)})\} = d(x_{i1}, x_{i2}) > 0; \tag{3.3}$$

$$d(x_1, x_2) \geq \frac{\sigma^2}{3}(1 + x_1^2 + x_2^2); \tag{3.4}$$

где $\varepsilon^{(i)}$ - случайные, неравноточные и некоррелированные ошибки с дисперсией $d(x_{i1}, x_{i2})$. $-1 \leq x_{ij} \leq 1, i = \overline{1, N}, j = \overline{1, 2}$.
Введём обозначения для точек:

$$x^{(1)} = (1, 1), x^{(2)} = (-1, 1), x^{(3)} = (-1, -1), x^{(4)} = (1, -1). \tag{3.5}$$

И для значений дисперсии наблюдений в этих точках:

$$d(x^{(1)}) = d_1, d(x^{(2)}) = d_2, d(x^{(3)}) = d_3, d(x^{(4)}) = d_4. \tag{3.6}$$

Согласно теореме 2, для указанной модели план

$$\varepsilon_1^0 = \begin{Bmatrix} x^{(1)}, x^{(2)}, x^{(3)} \\ \frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{3} \end{Bmatrix} \tag{3.7}$$

является точным D-оптимальным, если

$$d(x_1, x_2) \geq \frac{1}{4}(d_1 + d_3 + 2d_1x_1 - 2d_3x_2 - 2d_2x_1x_2 + (d_1 + d_2)x_1^2 + (d_2 + d_3)x_2^2). \quad (3.8)$$

Далее в этом разделе рассмотрим случай, когда неравенство 3.9 обращается в равенство:

$$d(x_1, x_2) = \frac{1}{4}(d_1 + d_3 + 2d_1x_1 - 2d_3x_2 - 2d_2x_1x_2 + (d_1 + d_2)x_1^2 + (d_2 + d_3)x_2^2). \quad (3.9)$$

3.1.2 Взвешенный метод наименьших квадратов для оценивания параметров

Для оценивания параметров регрессионной модели в условиях теоремы Гаусса-Маркова применяет метод наименьших квадратов (МНК). Неравноточность наблюдений нарушает условия теоремы Гаусса-Маркова, поэтому для оценки параметров неравноточной регрессионной модели применяется другой метод – взвешенный метод наименьших квадратов (взвешенный МНК). Взвешенный МНК является обобщением МНК, при котором каждое наблюдение учитывается с весом, обратно пропорциональным его дисперсии.

Обозначим X - матрица плана эксперимента, $y = (y_1, ..., y_N)^T$ – вектор наблюдений, $\varepsilon = (\varepsilon_1, ..., \varepsilon_N)^T$ – вектор случайных ошибок, $\theta = (\theta_0, \theta_1, \theta_2)^T$ – вектор параметров. Тогда (3.1) можно переписать в матричной форме:

$$y = X\theta + \varepsilon. \quad (3.10)$$

Обозначим $w = (w_1, ..., w_N)^T$ – вектор весов наблюдений, где $w_i = D(x_{i1}, x_{i2}) = d(x_{i1}, x_{i2})$. Также обозначим $W = diag(w)$ – диагональная матрица весов. Тогда $\hat{\theta}$ – оценка вектора параметров θ по взвешенному МНК имеет вид:

$$\hat{\theta} = (X^T W X)^{-1} X^T W y. \quad (3.11)$$

Обоснование и свойства взвешенного МНК могут быть найдены, например, в [6].

3.1.3 Численный эксперимент

Проведём численный эксперимент, чтобы понять, при каких условиях и на сколько D-оптимальный план эффективнее другого, случайного плана.

Алгоритм проведения эксперимента.

1. задать коэффициенты $\theta_0, \theta_1, \theta_2$ уравнения регрессии (3.1), значения d_1, d_2, d_3 и, соответственно, функцию $d(x_1, x_2)$ согласно (3.9);
2. задать количество наблюдений N, где N кратно 3;
3. сгенерировать матрицу плана эксперимента X одним из способов:
   (a) $X = X_{optimal}$ – согласно теореме о D-оптимальном плане, т.е. $N/3$ наблюдений в каждой из точек $x^{(1)}, x^{(2)}, x^{(3)}$;
   (b) $X = X_{random}$ – N случайных точек из двумерного равномерного распределения $R^2[-1, 1]$;
4. для X сгенерировать вектор ошибок ε с дисперсией $d(x_1, x_2)$;
5. вычислить $y = X\theta + \varepsilon$;
6. по взвешенному методу наименьших квадратов найти $\hat{\theta}$ – оценку вектора параметров θ;
7. измерить точность оценки как среднеквадратическую ошибку: $e = \frac{1}{3}\sum_{i=0}^{2}(\theta_i - \hat{\theta}_i)^2$.

Чтобы сделать результаты эксперимента более точными, описанный алгоритм повторяется T раз и в качестве итоговой оценки точности берётся $\overline{e} = \frac{1}{T}\sum_{t=1}^{T} e_t$.

Параметры эксперимента.

Зададим конкретные значения параметров для проведения эксперимента.

1. $\theta = (2, 5, 1)^T$;
2. $d_1 = 6, d_2 = 4, d_3 = 2$;
3. $N \in \{3, 6, ..., 24, 27\}$;
4. $T = 5$.

Реализация эксперимента.

Для реализация описанного эксперимента была написана программа на языке программирования *Python* с использованием пакетов *NumPy*, *SciPy*, *scikit-learn* [8]. Исходный код программы приведён в приложении Д

Результаты эксперимента.

Рассмотрим график 3.1 изменения средней квадратичной ошибки оценок в зависимости от числа проведённых наблюдений. Закрашенная область соответствует $\overline{e}$ с учётом стандартного отклонения $\overline{r}$. Синяя линяя соответствует оптимальному плану, оранжевая – случайному.

Видно, что D-оптимальный план эксперимента позволяет точно оценить па-

раметры модели уже при $N = 3$, в то время как случайный план является крайне неточными при $N < 9$. При $N > 9$ оба подхода дают почти одинаково точный результат.

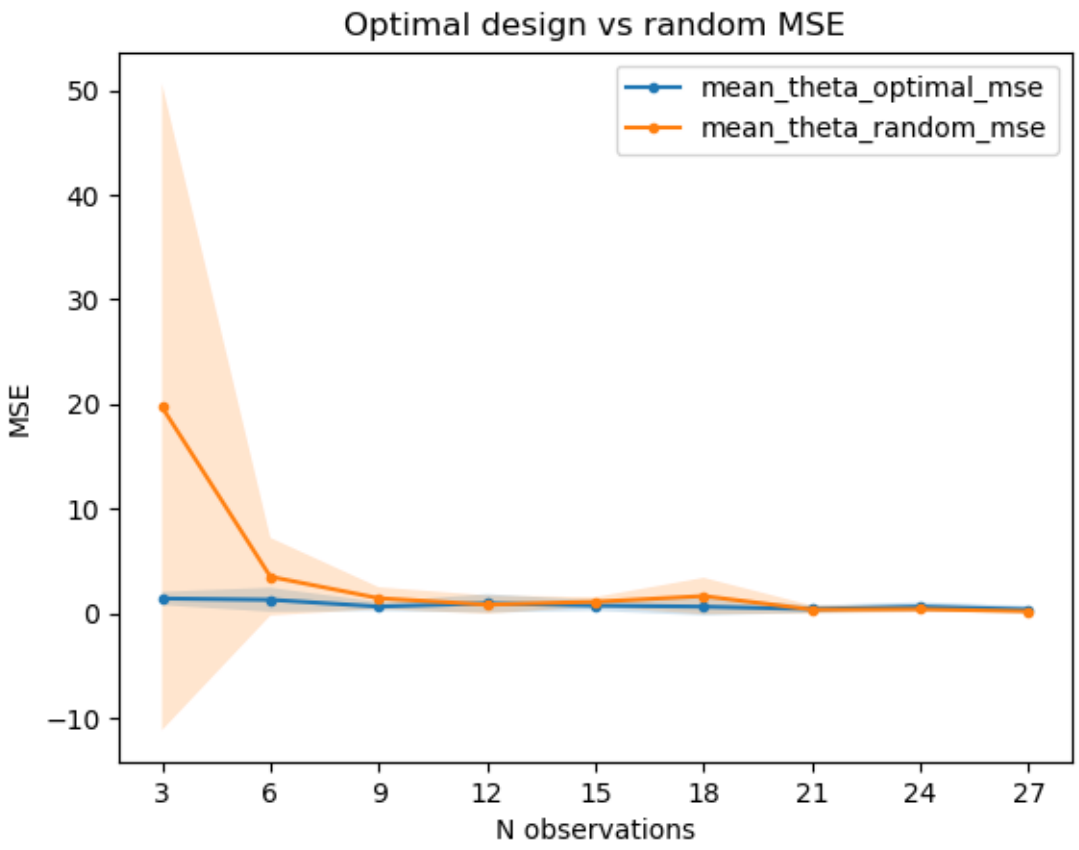

Рис. 3.1: График точности оценки параметров в зависимости от числа наблюдений

Таким образом D-оптимальный план позволяет существенно снизить число проводимых наблюдений при достижении той же точности оценок.

ЗАКЛЮЧЕНИЕ

В данной работе были построены непрерывные D-оптимальные планы для линейной множественной регрессии с неравноточными наблюдениями для случая, когда число неизвестных параметров равно 3; показано, что построенные планы также являются точными D-оптимальными планами; была показана невозможность существования точного D-оптимального плана во всех точках спектра для модели с 2-мя независимыми переменными и 3-мя параметрами; были получены результаты относительно размещения наблюдений для модели с линейным изменением. Также были построены непрерывные D-оптимальные планы для модели неравноточных наблюдений с 3-мя независимыми переменными и 3-мя параметрами. Экспериментальным путём была показана эффективность D-оптимальных планов при малом числе наблюдений.

Полученные результаты могут быть расширены на более широкий класс моделей.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Теория оптимального эксперимента (планирование регрессионных экспериментов). Федоров В.В., монография, Главная редакция физико-математической литературы изд-ва ”Наука”, 1971.

2. Stéfan van der Walt, S. Chris Colbert and Gaël Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation // Computing in Science & Engineering. – 2011. Vol. 13. – P. 22–30.

3. Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E.A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python // Nature Methods, in press. – 2020.

4. В.П. Кирлица – ”Точные D-оптимальные планы экспериментов для линейной множественной регрессии с неравноточными наблюдениями” // Журнал Белорусского государственного университета. Математика, информатика, 2017(3), с. 53-59.

5. В.П. Кирлица – ”Построение D-оптимальных планов экспериментов для линейной множественной регрессии с неравноточными наблюдениями” // Журнал Белорусского государственного университета. Математика, информатика, 2019(2), с. 27-33.

6. Айвазян С. А. Прикладная статистика. Основы эконометрики. Том 2. — М.: Юнити-Дана, 2001. — 432 с. — ISBN 5-238-00305-6.

7. Meurer A, Smith CP, Paprocki M, Čertík O, Kirpichev SB, Rocklin M, Kumar A, Ivanov S, Moore JK, Singh S, Rathnayake T, Vig S, Granger BE, Muller RP, Bonazzi F, Gupta H, Vats S, Johansson F, Pedregosa F, Curry MJ, Terrel AR, Roučka Š, Saboo A, Fernando I, Kulal S, Cimrman R, Scopatz A. SymPy: symbolic computing in Python. // PeerJ Computer Science. – 2017. Vol. 3. – 3:e103.

8. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay. Scikit-learn: Machine Learning in Python // Journal of Machine Learning Research. – 2011. Vol. 12. – P. 2825–2830.

ПРИЛОЖЕНИЕ А

Численная проверка оптимальности плана методом перебора точек единичного квадрата

```
import math
import numpy as np

x_from = -1
x_to = 1
h = 0.1
n = int((x_to - x_from) / h)

def d_134(x1, x2):
  return 2 - x1 + 3*x2 - 2*x1*x2 +
    1.5*x1**2 + 2.5*x2**2

def d_234(x1, x2):
  return 2 + 2*x1 + 2*x2 +
    x1*x2 +1.5*(x1**2) + 1.5*(x2**2)

def d_123(x1, x2):
  return 2 + 3*x1 - x2 - 2*x1*x2 +
    2.5*x1**2 + 1.5*x2**2

def d_124(x1, x2):
  return 2 - x1 + 3*x2 - 2*x1*x2 +
    1.5*x1^2 + 2.5*x2^2

d = d_134

def make_point(i, j):
  return (x_from + i*h, x_from + j*h)

def M(x):
```

```
31    matrix = np.zeros((3, 3))
32    for i in range(3):
33    v = [[1, x[i][0], x[i][1]]]
34    d_i = d(x[i][0], x[i][1])
35    matrix += (1/d_i) * np.matmul(np.transpose(v), v)
36    return matrix
37
38  def test(d):
39    max_points = []
40    max_M_det = float('-inf')
41    eps = 10e-6
42
43    for i1 in range(0, n + 1):
44      for j1 in range(0, n + 1):
45        for i2 in range(0, n + 1):
46          for j2 in range(0, n + 1):
47            for i3 in range(0, n + 1):
48              for j3 in range(0, n + 1):
49                x1 = make_point(i1, j1)
50                x2 = make_point(i2, j2)
51                x3 = make_point(i3, j3)
52
53                M_det=abs(
54                  np.linalg.det(M([x1, x2, x3]))
55                )
56                if M_det > max_M_det:
57                  max_M_det = M_det
58                  max_points = [x1, x2, x3]
59
60      print(max_points, max_M_det)
61
62    print('Plan 1, 2, 3')
63    test(d_123)
64    print('Plan 1, 2, 4')
```

```
65    test(d_124)
66    print('Plan 1, 3, 4')
67    test(d_134)
68    print('Plan 2, 3, 4')
69    test(d_234)
```

Листинг 1: Численная проверка оптимальности плана методом перебора точек единичного квадрата

ПРИЛОЖЕНИЕ Б

Символьная проверка оптимальности плана

```
disp('Plan in points (x2, x3 ,x4)')
disp(['d(-1, 1)=', char(check_234(-1, 1))])
disp(['d(-1, -1)=', char(check_234(-1, -1))])
disp(['d(1, -1)=', char(check_234(1, -1))])

disp('Plan in points (x1, x2 , x3)')
disp(['d(1, 1)=', char(check_123(1, 1))])
disp(['d(-1, 1)=', char(check_123(-1, 1))])
disp(['d(-1, -1)=', char(check_123(-1, -1))])

disp('Plan in points (x1, x2 , x3)')
disp(['d(1, 1)=', char(check_124(1, 1))])
disp(['d(-1, 1)=', char(check_124(-1, 1))])
disp(['d(1, -1)=', char(check_124(1, -1))])

disp('Plan in points (x1, x3 , x4)')
disp(['d(1, 1)=', char(check_134(1, 1))])
disp(['d(-1, -1)=', char(check_134(-1, -1))])
disp(['d(1, -1)=', char(check_134(1, -1))])



function res = check(a, b, c, e, x1, x2)
    num = a^2 - e^2 + 2*(c*e -a*b)*x1 + ...
      2*(b*e - a*c)*x2 + ...
      2*(b*c - a*e)*x1*x2 + ...
      (a^2 - c^2)*x1^2 + (a^2 - b^2)*x2^2;
    denum = a^3 + 2*b*c*e - a*(b^2 + c^2 + e^2);
    res = simplify(num / denum);
end

```

```
function res = check_234(x1, x2)
    syms d2 d3 d4 l2 l3 l4 a b c e;

    l2 = 1/d2;
    l3 = 1/d3;
    l4 = 1/d4;

    a = l2 + l3 + l4;
    b = -l2 -l3 + l4;
    c = l2 - l3 - l4;
    e = -l2 + l3 - l4;

    res = check(a, b, c, e, x1, x2);
end

function res = check_123(x1, x2)
    syms d1 d2 d3 l1 l2 l3 a b c e;

    l1 = 1/d1;
    l2 = 1/d2;
    l3 = 1/d3;

    a = l1 + l2 + l3;
    b = l1 - l2 - l3;
    c = l1 + l2 - l3;
    e = l1 - l2 + l3;

    res = check(a, b, c, e, x1, x2);
end

function res = check_124(x1, x2)
    syms d1 d2 d4 l1 l2 l4 a b c e;

    l1 = 1/d1;
```

```
66      l2 = 1/d2;
67      l4 = 1/d4;
68
69      a = l1 + l2 + l4;
70      b = l1 - l2 + l4;
71      c = l1 + l2 - l4;
72      e = l1 - l2 - l4;
73
74      res = check(a, b, c, e, x1, x2);
75  end
76
77  function res = check_134(x1, x2)
78      syms d1 d3 d4 l1 l3 l4 a b c e;
79
80      l1 = 1/d1;
81      l3 = 1/d3;
82      l4 = 1/d4;
83
84      a = l1 + l3 + l4;
85      b = l1 - l3 + l4;
86      c = l1 - l3 - l4;
87      e = l1 + l3 - l4;
88
89      res = check(a, b, c, e, x1, x2);
90  end
```

Листинг 2: Символьная проверка оптимальности плана

ПРИЛОЖЕНИЕ В

Размещение наблюдений в точках спектра плана

```
x_points = [
      (1, 1),
      (-1, 1),
      (-1, -1),
      (1, -1)
]


def M_det(a, b, c, e):
    return a**3 + 2*b*c*e - a*(b**2 + c**2 + e**2)


def M_det_raw(n1, d1, n2, d2, n3, d3, n4, d4):
    l1 = n1 / d1
    l2 = n2 / d2
    l3 = n3 / d3
    l4 = n4 / d4
    a = l1 + l2 + l3 + l4
    b = l1 - l2 - l3 + l4
    c = l1 + l2 - l3 - l4
    e = l1 - l2 + l3 - l4
    return M_det(a, b, c, e)


def make_d_surface(a0, a1, a2):
    return lambda x1, x2: a0 + a1*x1 + a2*x2


def place_observations(d_surface, n):
    d_values = [
      d_surface(*x_point)
```

```
32          for x_point in x_points
33      ]
34
35
36      max_abs_M_det = -1
37      best_placements = []
38      for n1 in range(0, n):
39          for n2 in range(0, n):
40              for n3 in range(0, n):
41                  if not 1 <= n1 + n2 + n3 <= n:
42                      continue
43                  n4 = n - (n1 + n2 + n3)
44                  ns = [n1, n2, n3, n4]
45                  abs_M_det = abs(M_det_raw(
46                      n1, d_values[0],
47                      n2, d_values[1],
48                      n3, d_values[2],
49                      n4, d_values[3]
50                  ))
51                  if max_abs_M_det < abs_M_det:
52                      best_placements = [ns]
53                      max_abs_M_det = abs_M_det
54                  elif abs(max_abs_M_det-abs_M_det)<1e-4:
55                      best_placements.append(ns)
56
57      return best_placements
58
59  d_surfaces_params = [
60      (40, 0, 0),
61      (40, -1, 0),
62      (40, -4, 0),
63      (40, -8, 0),
64      (40, -12, 0),
65      (40, -30, 0),
```

```
66      (40, -39, 0),
67      (40, -39.5, 0)
68  ]
69
70  n = 5
71
72  print(f'n = {n}')
73  for d_surface_params in d_surfaces_params:
74      d_surface = make_d_surface(*d_surface_params)
75      best_placement = place_observations(d_surface, n)
76      print('d(x1, x2) >= {} + {}*x1 + {}*x2'
77          .format(*d_surface_params))
78      print('best placements: ', best_placement)
```

Листинг 3: Размещение наблюдений в точках спектра плана

ПРИЛОЖЕНИЕ Г

Символьная проверка D-оптимальности плана с тремя переменными

```
from sympy import *
import numpy as np


# points
x_points_arr = [
    [1, 1, 1],
    [1, -1, 1],
    [1, -1, -1],
    [1, 1, -1],
    [-1, 1, 1],
    [-1, -1, 1],
    [-1, -1, -1],
    [-1, 1, -1],
]
x_points = {
    i: x_point
    for i, x_point in zip(range(1, 9), x_points_arr)
}

# function to build a plan
def build_plan_in_points(points, points_id):
    points = np.array(points)

    N = 3
    n_points = 3

    d_names = [
        f'd{point_id}' for point_id in points_id
    ]
```

```
31      d1, d2, d3 = symbols(d_names)
32      d = Matrix([d1, d2, d3])
33      X = Matrix(MatrixSymbol('X', N, n_points)).T
34
35      M = Matrix.zeros(N)
36      for i in range(n_points):
37          M +=  X[:, i] * X[:, i].T / d[i]
38
39      a, b, c, e = symbols('a b c e')
40      x0, x1, x2 = symbols('x0 x1 x2')
41
42      M_abc = Matrix([
43          [a, b, c],
44          [b, a, e],
45          [c, e, a]
46      ]) / 3
47
48      M_abc_inv = M_abc.inverse_ADJ()
49
50      x = Matrix([x0, x1, x2])
51
52      df =  x.T * M_abc_inv * x / 3
53      df.simplify()
54
55      X_subs_map = {
56          X[i, j]: points[j, i]
57          for i in range(n_points)
58          for j in range(n_points)
59      }
60
61      M_exact = M.subs(X_subs_map)
62
63      abc_subs_map = {
64          'a': M_exact[0, 0],
```

```
65          'b': M_exact[0, 1],
66          'c': M_exact[0, 2],
67          'e': M_exact[1, 2]
68      }
69
70      df_exact = df.subs(abc_subs_map)
71      df_exact.simplify()
72      df_exact = df_exact[0]
73      df_exact = df_exact.collect(['x0', 'x1', 'x2'])
74
75      return df_exact
76
77
78  def prove_eq_in_points(d, points, points_id):
79      for point_id, point in zip(points_id, points):
80          subs_map = dict(zip(['x0', 'x1', 'x2'], point))
81          print(f'd(x{point_id}) =', d.subs(subs_map))
82
83
84  plans_points_ids = [
85      [1, 2, 3],
86      [1, 2, 4],
87      [1, 3, 4],
88      [2, 3, 4]
89  ]
90
91  if __name__ == '__main__':
92      for plan_id, points_id in enumerate(
          plans_points_ids):
93          plan_points = [
94              x_points[point_id]
95              for point_id in points_id
96          ]
97
```

```
98          d = build_plan_in_points(plan_points, points_id
                )
99
100         print(f'plan #{plan_id + 1}:')
101         print('d(x0, x1, x2) >= ', d)
102
103         print('prove that equal in points:')
104         prove_eq_in_points(d, plan_points, points_id)
105
106         print()
```

Листинг 4: Символьная проверка D-оптимальности плана с тремя переменными

ПРИЛОЖЕНИЕ Д

Сравнение качества оценок точного D-оптимального и случайного плана

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns

from sklearn.metrics import mean_squared_error
from sklearn.linear_model import LinearRegression

thetas = np.array([2, 5, 1])
def f(X):
    return np.dot(X, thetas)

d1, d2, d3 = 6, 4 , 2

def d_lower(x1, x2, d1, d2, d3):
    return 1/4 * (d1 + d3 + 2*d1*x2 - 2*d3*x2 - 2*d2*x2
        + (d1 + d2)*x1**2 + (d2 + d2)*x2**2)

points = np.array([
    [1, 1],
    [-1, 1],
    [-1, -1]
])

n_trials = 5
results = []

np.random.seed(42)

def make_y(X):
```

```
30      N = X.shape[0]
31      error_dispersions = d_lower(X[:, 1], X[:, 2], d1,
          d2, d3)
32      errors = np.random.normal(0, np.sqrt(
          error_dispersions), N)
33      return f(X) + errors
34
35  Ns = np.arange(1, 10) * 3
36  for _ in range(n_trials):
37      for N in Ns:
38          result = {
39              'N': N
40          }
41          Xs = [
42              ('optimal', np.repeat(points, N / 3, axis
                  =0)),
43              ('random', np.random.uniform(-1, 1, (N, 2))
                  )
44          ]
45          for name, X in Xs:
46              X = np.concatenate([np.ones((N, 1)), X],
                  axis=1)
47              y = make_y(X)
48              error_dispersions = d_lower(X[:, 1], X[:,
                  2], d1, d2, d3)
49              weights = 1 / error_dispersions
50              wls_estimator = LinearRegression(
                  fit_intercept=False).fit(X, y, weights)
51              result[f'theta_{name}_mse'] =
                  mean_squared_error(thetas, wls_estimator.
                  coef_)
52          results.append(result)
53
54  df = pd.DataFrame(results)
```

```
55
56 df_agg = df.groupby('N').agg(['mean', 'std'])
57 df_agg.columns = [f'{agg_name}_{column}' for (column,
     agg_name) in df_agg.columns]
58
59 ax = plt.gca()
60 for name in ['optimal', 'random']:
61     mean = df_agg[f'mean_theta_{name}_mse'].values
62     ax.plot(Ns, mean, '.-', label=f'mean_theta_{name}
       _mse')
63     mean = df_agg[f'mean_theta_{name}_mse']
64     std = df_agg[f'std_theta_{name}_mse']
65     ax.fill_between(Ns, mean - std, mean + std, alpha
       =0.2)
66 ax.legend()
67 ax.set_title('Optimal design vs random MSE')
68 ax.set_xticks(Ns)
69 ax.set_xticklabels(Ns)
70 ax.set_xlabel('N observations')
71 ax.set_ylabel('MSE')
72
73 plt.gcf().savefig('tex/images/heteroscedastic-
     experiment.png')
```

Листинг 5: Сравнение качества оценок точного D-оптимального и случайного плана