Санкт-Петербургский государственный университет
Прикладная математика и информатика
Вычислительная стохастика и статистические модели

Отчёт по задачам на тему «Моделирование распределений»

Выполнил:
Яковлев Денис Михайлович
Группа 21.Б04-мм
095998@student.spbu.ru

Санкт-Петербург
3 апреля 2024 г.

Оглавление

1. Задача. Формулировка

(MCB 309) *Промоделировать 3-мя способами распределение с плотностью:*

$$\rho(x) = C \begin{cases} 2 - x, & 0 < x \leqslant 1; \\ 1 + x, & 1 < x \leqslant 2; \\ 2e^{-x}, & x > 2; \\ 0, & \textit{иначе.} \end{cases}$$

1.1. Метод обратных функций

Найдём функцию распределения $F(x)$:

$$F(x) = C \begin{cases} 0, & x \leqslant 0; \\ 2x - \dfrac{x^2}{2}, & 0 < x \leqslant 1; \\ x + \dfrac{x^2}{2}, & 1 < x \leqslant 2; \\ 4 + 2e^{-2} - 2e^{-x}, & x > 2. \end{cases}$$

Определим, чему равна константа C:

$$\lim_{x \to +\infty} F(x) = C(4 + 2e^{-2}) = 1 \Rightarrow C = \frac{1}{4 + 2e^{-2}}.$$

При $y \in (0, 1)$ решим уравнение $y = F(x)$ относительно x, чтобы получить обратную функцию $x = F^{-1}(y)$.

Решим уравнение $C(2x - \dfrac{x^2}{2}) = y$ при условии, что $0 < x \leqslant 1$:

$$C\left(2x - \frac{x^2}{2}\right) = y \Leftrightarrow x_\pm = \frac{4 \pm 2\sqrt{2}\sqrt{2 - \dfrac{y}{C}}}{2}$$

$$\overset{x_+ \geqslant 2}{\Rightarrow} x_- = 2 - \sqrt{4 - \frac{2y}{C}} \Rightarrow y = C\left(2 - \frac{(4 - 2x_-)^2}{8}\right).$$

Подставляя $x_- = 0,\ x_- = 1$, получаем $y = 0,\ y = \dfrac{3}{2}C$ соответственно.

Далее решим $C(x + \dfrac{x^2}{2}) = y$ и получим:

$$x = -1 + \sqrt{1 + \frac{2y}{C}}.$$

При $x = 1,\ x = 2$ получим $y = \dfrac{3}{2}C,\ y = 4C$ соответственно.
Осталось решить $C(4 + 2e^{-2} - 2e^{-x}) = C(C^{-1} - 2e^{-x}) = y$ относительно x. Получим:

$$x = -\log\left(\frac{1-y}{2C}\right)$$

при $4C < y < 1$. Таким образом, обратная функция распределения:

$$F^{-1}(y) = \begin{cases} 2 - \sqrt{4 - \dfrac{2y}{C}}, & 0 < y \leqslant \dfrac{3}{2}C; \\ -1 + \sqrt{1 + \dfrac{2y}{C}}, & \dfrac{3}{2}C < y \leqslant 4C; \\ -\log\left(\dfrac{1-y}{2C}\right), & 4C < y < 1 \end{cases}$$

Algorithm 1 Метод обратных функций

```
Get(α); C ← 1/(4+2e^{-2})
d[0] ← (3/2)C; d[1] ← 4C
if α < d[0] then
    ξ ← 2 − √2 √(2 − α/C)
else if α < d[1] then
    ξ ← −1 + √(1 + 2α/C)
else
    ξ ← −log((1 − α)/(2C))
end if
```

1.2. Метод декомпозиции

Разложим плотность распределения

$$\rho(x) = C(2-x)\mathbb{1}_{(0,1]} + C(1+x)\mathbb{1}_{(1,2]} + 2Ce^{-x}\mathbb{1}_{(2,+\infty)}$$

и представим в виде смеси плотностей:

$$\rho_1(x) = \frac{2}{3}(2-x)\mathbb{1}_{(0,1]},$$
$$\rho_2(x) = \frac{2}{5}(1+x)\mathbb{1}_{(1,2]},$$
$$\rho_3(x) = \frac{1}{e^{-2}}\ e^{-x}\mathbb{1}_{(2,+\infty)}$$

Тогда представим плотность через смесь плотностей $\rho(x) = q_1\rho_1(x) + q_2\rho_2(x) + q_3\rho_3(x)$ такую, что:

$$q_1 = \frac{3}{2}C;$$
$$q_2 = \frac{5}{2}C;$$
$$q_3 = 2e^{-2}C.$$

При этом $q_1 + q_2 + q_3 = \dfrac{4+2e^{-2}}{4+2e^{-2}} = 1$. Для $\rho_1(x), \rho_2(x), \rho_3(x)$ найдём соответствующие им обратные функции распределения:

$$F_1(x) = \frac{2}{3}(2x - \frac{x^2}{2}),\ F_1^{-1}(y) = 2 - \sqrt{2}\sqrt{2 - \frac{3y}{2}};$$
$$F_2(x) = \frac{2}{5}(x + \frac{x^2}{2} - \frac{3}{2}),\ F_2^{-1}(y) = -1 + \sqrt{4+5y};$$
$$F_3(x) = \frac{e^{-2} - e^{-x}}{e^{-2}},\ F_3^{-1}(y) = 2 - \log(1-y),$$

где $y \in (0, 1)$.

Algorithm 2 Метод декомпозиции

Get(α_1, α_2); $C \leftarrow \frac{1}{4+2e^{-2}}$
$q[0] \leftarrow \frac{3}{2}C$; $q[1] \leftarrow 4C$
if $\alpha_1 < q[0]$ **then**
    $\xi \leftarrow 2 - \sqrt{4 - 3\alpha_2}$
else if $\alpha_1 < q[1]$ **then**
    $\xi \leftarrow -1 + \sqrt{4 + 5\alpha_2}$
else
    $\xi \leftarrow 2 - \log(1 - \alpha_2)$
end if

1.3. Метод отбора

В качестве мажорирующего распределения возьмём $q(x) = e^{-x}\mathbb{1}_{(0,+\infty)}$, $q \sim \mathrm{EXP}(1)$. Его распределение можно промоделировать следующим образом: $\eta = -\log(\alpha_1)$.

Производная Радона-Никодима $r(x)$:

$$r(x) = p(x)/q(x) = C \begin{cases} 0, & x \leqslant 0; \\ (2-x)e^x, & 0 < x \leqslant 1; \\ (1+x)e^x, & 1 < x \leqslant 2; \\ 2, & 2 < x. \end{cases}$$

$$M = 3Ce^2 \approx 5.19,\ C = (4 + 2e^{-2})^{-1} \approx 0.234,$$

где M — мажоранта. Так как проверяем неравенство $r(\eta) < M\alpha_2$, то как $r(x)$, так и M можно разделить на общий множитель C. Тогда $M = 3e^2 \approx 22.167$.

Далее, неравенство $r(\eta) < M\alpha_2$ выполняется тогда и только тогда, когда либо

$$0 < \eta \leqslant 1,\ (2-\eta)e^\eta < 3e^2\alpha_2$$

либо

$$1 < \eta \leqslant 2,\ (1+\eta)e^\eta < 3e^2\alpha_2$$

либо

$$2 < \eta,\ 2 < 3e^2\alpha_2$$

Откуда выходят следующие условия:

$$\alpha_2 > \frac{1}{3}(2-\eta)e^{\eta-2},\ 0 < \eta \leqslant 1; \tag{1}$$

$$\alpha_2 > \frac{1}{3}(1+\eta)e^{\eta-2},\ 1 < \eta \leqslant 2; \tag{2}$$

$$\alpha_2 > \frac{2}{3e^2},\ 2 < \eta; \tag{3}$$

Тогда $r(\eta) < M\alpha_2$ совпадает с этими условиями:

$$e^{-1} < \alpha_1,\ \alpha_2\alpha_1 > (2 + \log(\alpha_1))M^{-1} \tag{1}$$

$$e^{-2} < \alpha_1 \leqslant e^{-1},\ \alpha_2\alpha_1 > (1 - \log(\alpha_1))M^{-1} \tag{2}$$

$$0 < \alpha_1 \leqslant e^{-2},\ \alpha_2 > 2M^{-1} \tag{3}$$

Поскольку условия получились слишком объёмными, в описанном ниже алгоритме рассматривается случай $r(\eta) \geqslant M\alpha_2$, что соответствует следующим условиям:

$$e^{-1} < \alpha_1,\ \alpha_2\alpha_1 \leqslant (2 + \log(\alpha_1))M^{-1}$$

$$e^{-2} < \alpha_1 \leqslant e^{-1},\ \alpha_2\alpha_1 \leqslant (1 - \log(\alpha_1))M^{-1}$$

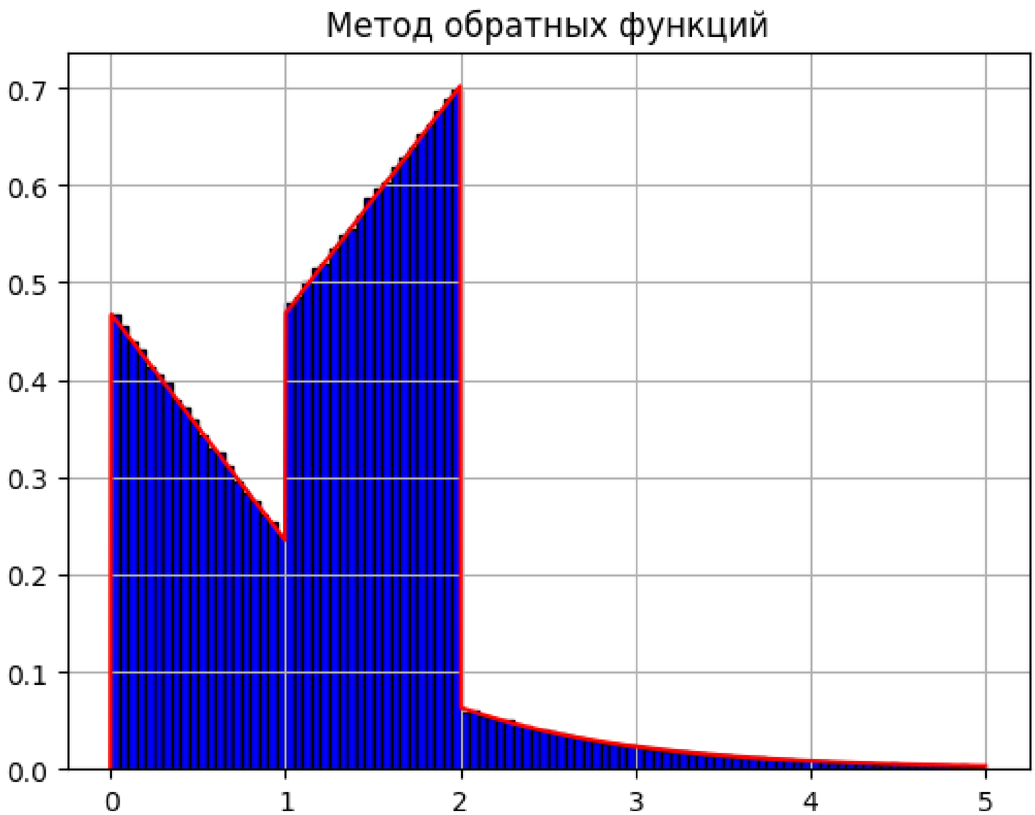
$$0 < \alpha_1 \leqslant e^{-2},\ \alpha_2 \leqslant 2M^{-1}$$

Algorithm 3 Метод отбора

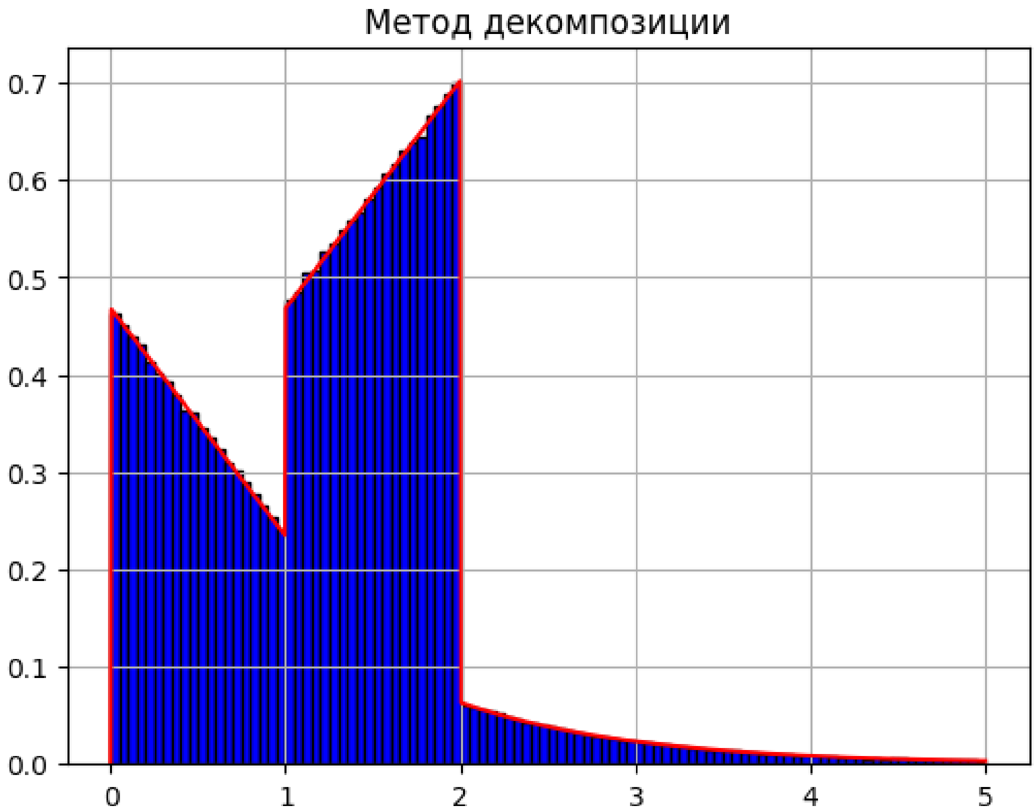
$M_1 \leftarrow \frac{1}{3e^2};\ M_2 \leftarrow 2M_1;\ b_1 \leftarrow e^{-1};\ b_2 \leftarrow b_1 * b_1$
while TRUE **do**
    Get(α_1, α_2)
    if $b_1 < \alpha_1$ and $\alpha_2\alpha_1 \leqslant (2 + \log(\alpha_1))M_1$ **then**
        $\xi \leftarrow -\log(\alpha_1)$
    end if
    if $b_2 < \alpha_1 \leqslant b_1$ and $\alpha_1\alpha_2 \leqslant (1 - \log(\alpha_1))M_1$ **then**
        $\xi \leftarrow -\log(\alpha_1)$
        break
    end if
    if $\alpha_1 \leqslant b_2$ and $\alpha_2 \leqslant M_2$ **then**
        $\xi \leftarrow -\log(\alpha_1)$
        break
    end if
end while

2. Реализации алгоритмов

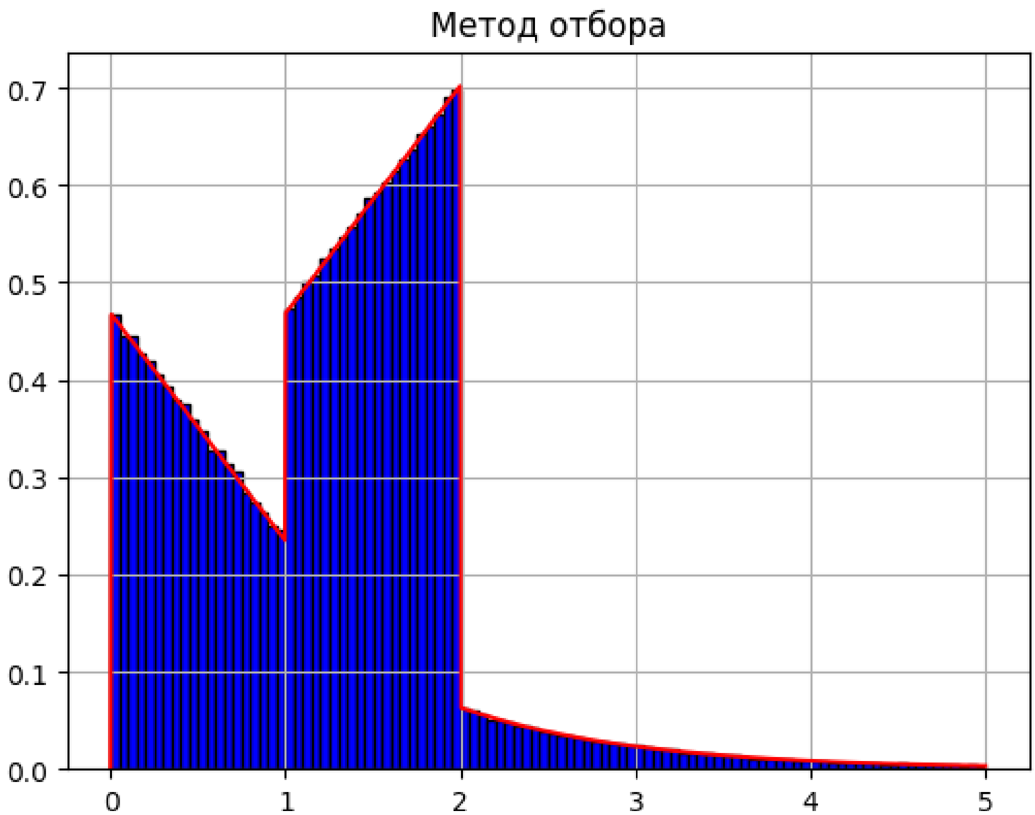
```python
import math
from random import random
import matplotlib.pyplot as plt
import numpy as np


n = 10**6
c = 1/(4+2*math.exp(-2))

def p(x):
    if(0<x and x<=1):
        return c*(2-x)
    elif(1 < x and x <= 2):
        return c*(1+x)
    elif 2<x:
        return 2*c*math.exp(-x)
    else:
        return 0

def Plot_Histogram(data, title):
    plt.hist(data, bins=100, density=True, range=(0,5), color='blue',
edgecolor='black')
    plt.title("Гистограмма")
    x_values = np.linspace(0, 5, 1000)
    y_values = [p(x) for x in x_values]
    plt.plot(x_values, y_values, color = 'red')
    plt.grid(True)
    plt.title(title)
    plt.show()

def Inverse_Transform_Method(n):
    d = [3/2*c, 4*c]
    ci = c**(-1)
    xi = []
    for i in range(n):
        alpha = random()
        if(alpha < d[0]):
            xi.append(2-math.sqrt(4-2*alpha*ci))
        elif(alpha < d[1]):
            xi.append(-1 + math.sqrt(1+2*alpha*ci))
        else:
            xi.append(-math.log((1-alpha)*ci/2))
    return xi

xi = Inverse_Transform_Method(n)
Plot_Histogram(xi, "Метод обратных функций")
```




```
def Decomposition_Method(n):
    q = [3/2*c, 4*c]
    xi = []
    for i in range(n):
        alpha_1, alpha_2 = random(), random()
        if(alpha_1 < q[0]):
            xi.append(2-math.sqrt(4-3*alpha_2))
        elif(alpha_1 < q[1]):
            xi.append(-1 + math.sqrt(4+5*alpha_2))
        else:
            xi.append(2 - math.log(1-alpha_2))
    return xi

xi = Decomposition_Method(n)
Plot_Histogram(xi, "Метод декомпозиции")
```




```
def Selection_Method(n):
    M_1 = 1/(3*math.exp(2))
    M_2 = 2*M_1
    b_1 = math.exp(-1)
    b_2 = b_1*b_1
    xi = []
    for i in range(n):
        while 1:
            alpha_1, alpha_2 = random(), random()
            if (b_1 < alpha_1) and (alpha_2*alpha_1 <
(2+math.log(alpha_1))*M_1):
                xi.append(-math.log(alpha_1))
                break
            if (b_2 < alpha_1 <= b_1) and (alpha_1*alpha_2 < (1-
math.log(alpha_1))*M_1):
                 xi.append(-math.log(alpha_1))
                 break
            if (alpha_1 <= b_2) and (alpha_2 < M_2):
                 xi.append(-math.log(alpha_1))
                 break
    return xi

xi = Selection_Method(n)
Plot_Histogram(xi, "Метод отбора")
```


Метод отбора
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
1
2
3
4
5