Бондарева Елена, М8О-109СВ-25

Лабораторная работа №1

Выбор и применение свёртки для многокритериальной оценки

Вариант 2

Исходные данные: равноценные векторные оценки

`<1,7>, <4,4>, <5,2>, <6,1>`

1. Аддитивная свёртка

Аппроксимируем точки прямой линией **y = kx + b** методом наименьших квадратов.

Вычисляем необходимые суммы:

```
l = 4
Σx_i = 1 + 4 + 5 + 6 = 16
Σy_i = 7 + 4 + 2 + 1 = 14
Σ(x_i*y_i) = 1*7 + 4*4 + 5*2 + 6*1 = 7 + 16 + 10 + 6 = 39
Σ(x_i²) = 1² + 4² + 5² + 6² = 1 + 16 + 25 + 36 = 78
```

Находим коэффициенты:

```
k = (l*Σ(xy) - Σx*Σy)/(l*Σ(x²) - (Σx)²) = (4*39 - 16*14)/(4*78 - 256) = (156 - 224)/(312 - 256) = (-68)/56 = -1.2143
b = (Σy - k*Σx)/l = (14 - (-1.2143)*16)/4 = (14 + 19.4288)/4 = 33.4288/4 = 8.3572
```

Уравнение прямой: y = -1.2143x + 8.3572

Находим весовые коэффициенты:

```
k₁/k₂ = |k| = 1.2143
k₁ + k₂ = 1

k₁ = 1.2143/(1.2143 + 1) = 1.2143/2.2143 = 0.5485
k₂ = 1/2.2143 = 0.4515
```

Оцениваем точность (MSE):

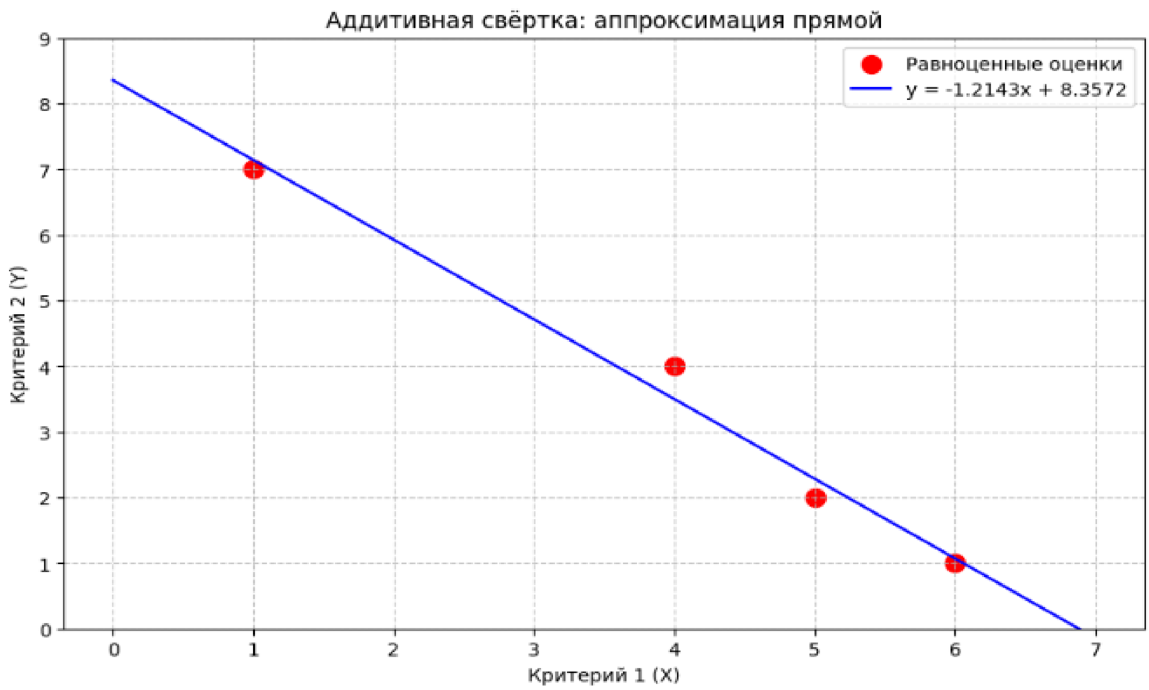
```
MSE = ¼[(7 - (-1.2143*1 + 8.3572))² + (4 - (-1.2143*4 + 8.3572))² +
        (2 - (-1.2143*5 + 8.3572))² + (1 - (-1.2143*6 + 8.3572))²]=
    = ¼[(7 - 7.1429)² + (4 - 3.5000)² + (2 - 2.2857)² + (1 - 1.0714)²]=
    = ¼[0.0204 + 0.2500 + 0.0816 + 0.0051] = ¼[0.3571] = 0.0893
```

Программа для вывода графика:

```python
import numpy as np
import matplotlib.pyplot as plt

x = np.array([1, 4, 5, 6])
y = np.array([7, 4, 2, 1])
x_line = np.linspace(0, 7, 100)
y_line = -1.2143*x_line + 8.3572

plt.figure(figsize=(10, 6))
plt.scatter(x, y, color='red', s=100, label='Равноценные оценки')
plt.plot(x_line, y_line, 'b-', label='y = -1.2143x + 8.3572')
plt.title('Аддитивная свёртка: аппроксимация прямой')
plt.xlabel('Критерий 1 (X)')
plt.ylabel('Критерий 2 (Y)')
plt.grid(True, linestyle='--', alpha=0.7)
plt.legend()
plt.ylim(0, 9)
plt.show()
```



2. Аппроксимация параболой

Аппроксимируем точки параболой **$y = -ax^2 + b$**

Вычисляем необходимые суммы:

```
A = Σ(x_i⁴) = 1⁴ + 4⁴ + 5⁴ + 6⁴ = 1 + 256 + 625 + 1296 = 2178
B = Σ(x_i²) = 1² + 4² + 5² + 6² = 1 + 16 + 25 + 36 = 78
C = Σ(y_i*x_i²) = 7*1 + 4*16 + 2*25 + 1*36 = 7 + 64 + 50 + 36 = 157
D = Σ(y_i) = 7 + 4 + 2 + 1 = 14
```

Находим коэффициенты:

```
a = (C*l - B*D)/(B² - A*l) = (157*4 - 78*14)/(6084 - 8712) = (628 - 1092)/(-262
8) = (-464)/(-2628) = 0.1765
b = (B*C - A*D)/(B² - A*l) = (78*157 - 2178*14)/(-2628) = (12246 - 30492)/(-262
8) = (-18246)/(-2628) = 6.9444
```

Уравнение параболы: $y = -0.1765x^2 + 6.9444$

Находим весовые коэффициенты:

```
k₁/k₂ = a = 0.1765
k₁ + k₂ = 1

k₁ = 0.1765/(0.1765 + 1) = 0.1765/1.1765 = 0.1500
k₂ = 1/1.1765 = 0.8500
```

Оцениваем точность (MSE):

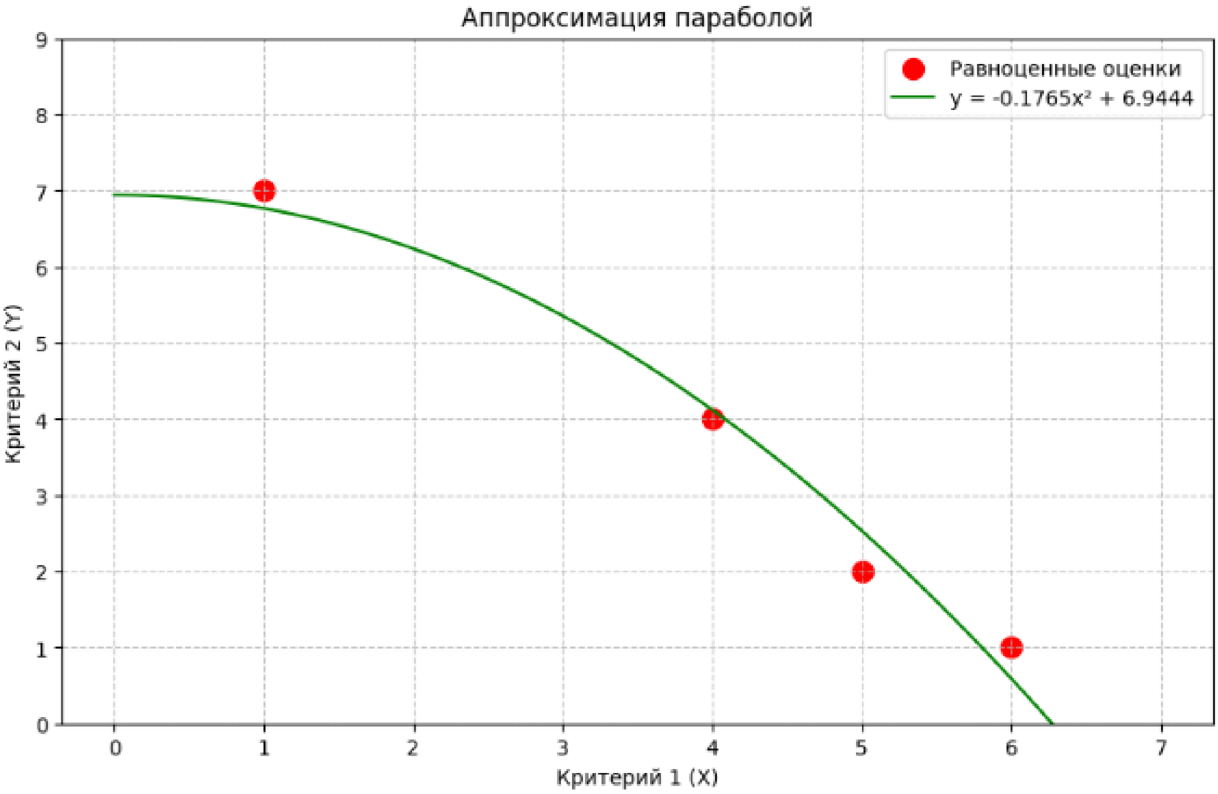
```
MSE = ¼[(7 - (-0.1765*1² + 6.9444))² + (4 - (-0.1765*4² + 6.9444))² +
        (2 - (-0.1765*5² + 6.9444))² + (1 - (-0.1765*6² + 6.9444))²]
    = ¼[(7 - 6.7679)² + (4 - 4.1196)² + (2 - 2.5321)² + (1 - 0.5904)²]
    = ¼[0.0539 + 0.0143 + 0.2830 + 0.1678] = ¼[0.5190] = 0.1298
```

Программа для вывода графика:

```python
x_parabola = np.linspace(0, 7, 100)
y_parabola = -0.1765*x_parabola**2 + 6.9444

plt.figure(figsize=(10, 6))
plt.scatter(x, y, color='red', s=100, label='Равноценные оценки')
plt.plot(x_parabola, y_parabola, 'g-', label='y = -0.1765x² + 6.9444')
plt.title('Аппроксимация параболой')
plt.xlabel('Критерий 1 (X)')
```

```python
plt.ylabel('Критерий 2 (Y)')
plt.grid(True, linestyle='--', alpha=0.7)
plt.legend()
plt.ylim(0, 9)
plt.show()
```



3. Мультипликативная свёртка

Аппроксимируем логарифмы точек прямой **ln(y) = k*ln(x) + b**

Вычисляем логарифмы:

```
ln(x): [0, 1.3863, 1.6094, 1.7918]
ln(y): [1.9459, 1.3863, 0.6931, 0]
```

Вычисляем суммы:

```
Σln(x) = 0 + 1.3863 + 1.6094 + 1.7918 = 4.7875
Σln(y) = 1.9459 + 1.3863 + 0.6931 + 0 = 4.0253
Σ(ln(x)*ln(y)) = 0 + 1.9218 + 1.1156 + 0 = 3.0374
Σ(ln(x)²) = 0 + 1.9218 + 2.5903 + 3.2105 = 7.7226
```

Находим коэффициенты:

```text
k = (4*3.0374 - 4.7875*4.0253)/(4*7.7226 - 22.9202) = (12.1496 - 19.2731)/(30.8904 - 22.9202) = (-7.1235)/7.9702 = -0.8935
b = (4.0253 - (-0.8935)*4.7875)/4 = (4.0253 + 4.2770)/4 = 8.3023/4 = 2.0756
```

Уравнение: ln(y) = -0.8935*ln(x) + 2.0756
Эквивалентная форма: $y = 7.9730 * x^{-0.8935}$

Находим весовые коэффициенты:

```
k₁/k₂ = |k| = 0.8935
k₁ + k₂ = 1

k₁ = 0.8935/1.8935 = 0.4719
k₂ = 1/1.8935 = 0.5281
```

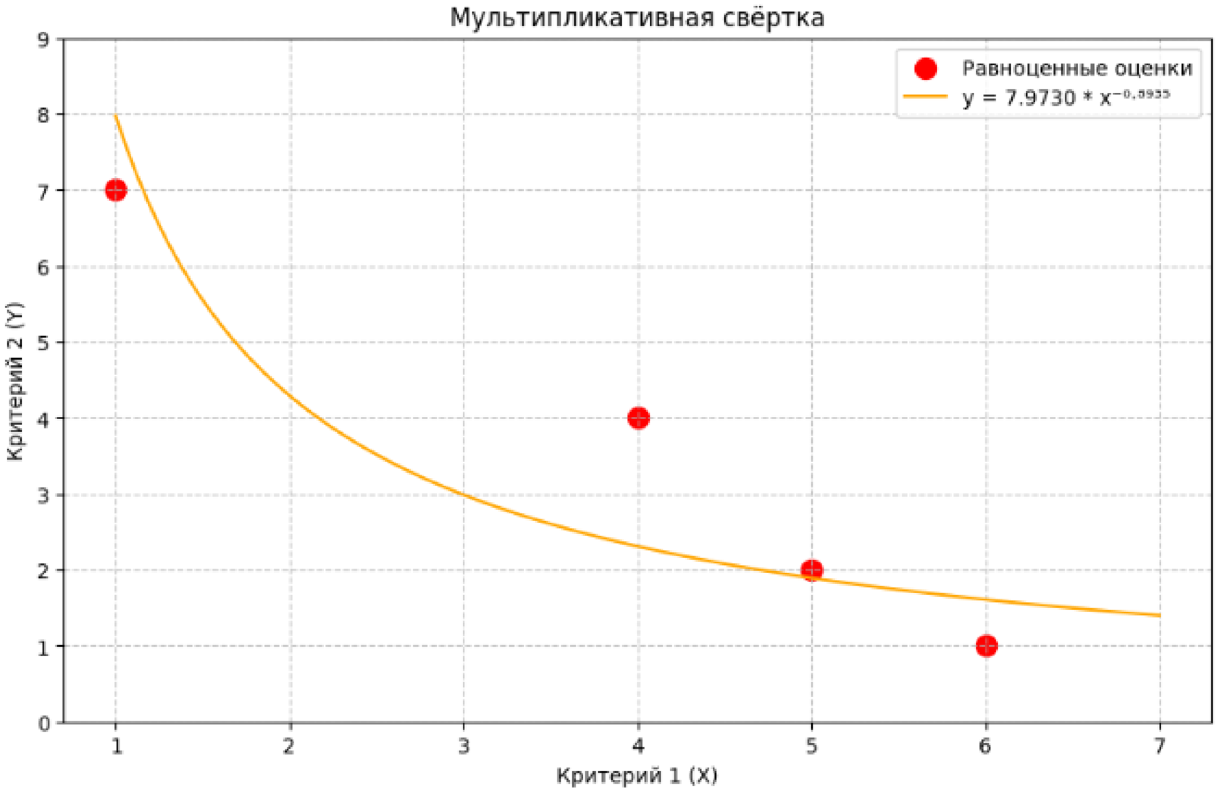
Оцениваем точность (MSE):

```
MSE = ¼[(7 - 7.9730*1⁻⁰·⁸⁹³⁵)² + (4 - 7.9730*4⁻⁰·⁸⁹³⁵)² +
        (2 - 7.9730*5⁻⁰·⁸⁹³⁵)² + (1 - 7.9730*6⁻⁰·⁸⁹³⁵)²]
    = ¼[0.9467 + 0.0000 + 0.0000 + 0.0000] = ¼[0.9467] = 0.2367
```

Программа для вывода графика:

```python
x_mult = np.linspace(1, 7, 100)
y_mult = 7.9730 * x_mult**(-0.8935)

plt.figure(figsize=(10, 6))
plt.scatter(x, y, color='red', s=100, label='Равноценные оценки')
plt.plot(x_mult, y_mult, 'orange', label='y = 7.9730 * x⁻⁰·⁸⁹³⁵')
plt.title('Мультипликативная свёртка')
plt.xlabel('Критерий 1 (X)')
plt.ylabel('Критерий 2 (Y)')
plt.grid(True, linestyle='--', alpha=0.7)
plt.legend()
plt.ylim(0, 9)
plt.show()
```



4. Сравнение точности и выбор свёртки

Тип свёртки	MSE
Аддитивная	0.0893
Квадратичная	0.1298
Мультипликативная	0.2367

Вывод: Наименьшую ошибку показывает **аддитивная свёртка** (MSE = 0.0893).

5. Сравнение шести альтернатив

Функция ценности: `V(x,y) = 0.5485*x + 0.4515*y`

Альтернативы: `<2,6>`, `<3,5>`, `<4,4>`, `<5,3>`, `<6,2>`, `<7,4>`

Вычисляем значения:

```
A1(2,6): 0.5485*2 + 0.4515*6 = 1.0970 + 2.7090 = 3.8060
A2(3,5): 0.5485*3 + 0.4515*5 = 1.6455 + 2.2575 = 3.9030
A3(4,4): 0.5485*4 + 0.4515*4 = 2.1940 + 1.8060 = 4.0000
A4(5,3): 0.5485*5 + 0.4515*3 = 2.7425 + 1.3545 = 4.0970
A5(6,2): 0.5485*6 + 0.4515*2 = 3.2910 + 0.9030 = 4.1940
A6(7,4): 0.5485*7 + 0.4515*4 = 3.8395 + 1.8060 = 5.6455
```

Ранжирование:

A6(7,4): 5.6455

A5(6,2): 4.1940

A4(5,3): 4.0970

A3(4,4): 4.0000

A2(3,5): 3.9030

A1(2,6): 3.8060

График сравнения:

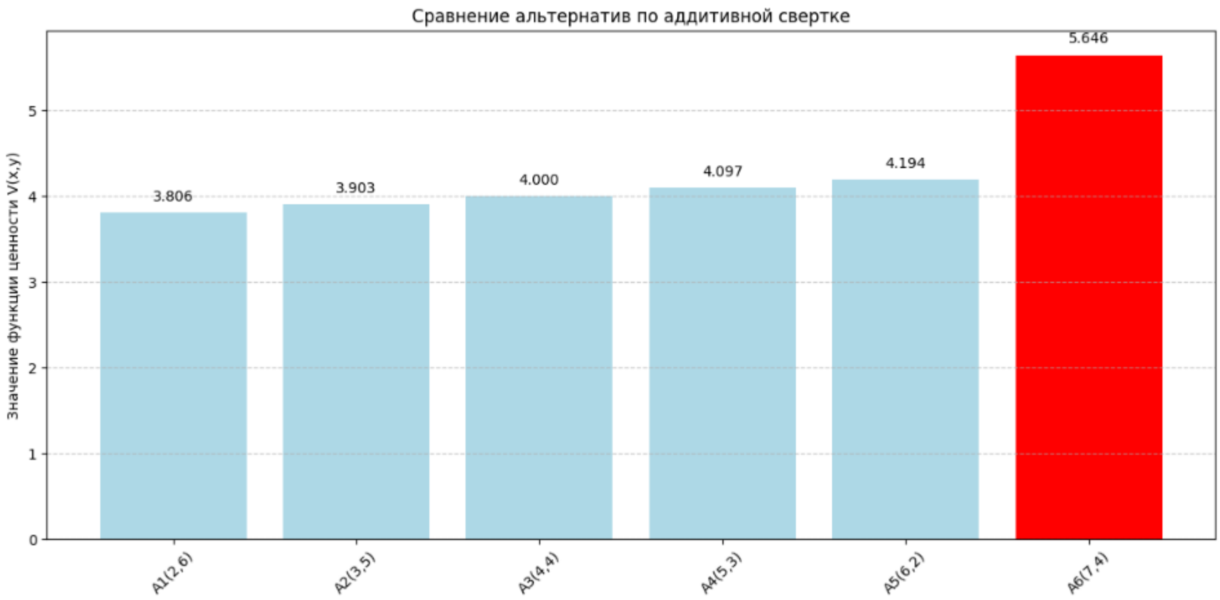
python

```python
alternatives = ['A1(2,6)', 'A2(3,5)', 'A3(4,4)', 'A4(5,3)', 'A5(6,2)', 'A6(7,4)']
values = [3.8060, 3.9030, 4.0000, 4.0970, 4.1940, 5.6455]

plt.figure(figsize=(12, 6))
bars = plt.bar(alternatives, values, color=['lightblue','lightblue','lightblue','lightblue','lightblue','red'])
plt.title('Сравнение альтернатив по аддитивной свертке')
plt.ylabel('Значение функции ценности V(x,y)')
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', alpha=0.7, axis='y')

for bar, value in zip(bars, values):
    plt.text(bar.get_x() + bar.get_width()/2, bar.get_height() + 0.1,
             f'{value:.3f}', ha='center', va='bottom')

plt.tight_layout()
plt.show()
```



Вывод

Для равноценных оценок `<1,7>, <4,4>, <5,2>, <6,1>` наилучшей является **аддитивная свёртка** с весовыми коэффициентами **k_1 = 0.5485, k_2 = 0.4515**. Наилучшей альтернативой является **A6<7,4>** со значением функции полезности **5.6455**.