Лосева Елизавета Юрьевна, группа 8-2

Лабораторная работа № 5

Вариант № 2

Распознавание образов, описываемых бинарными признаками

Цель работы

Исследовать алгоритмы оценивания плотности распределения случайных величин и случайных векторов на основе методов Парзена и k ближайших соседей.

Задание

Вычислить среднеквадратичную ошибку оценивания плотности распределения случайной величины по методу Парзена для окна вида:

a. $$\frac{1}{h}\phi\left(\frac{x-x^{(i)}}{h}\right)=\frac{1}{\pi h}\frac{1}{1+\left((x-x^{(i)})/h\right)^2};$$

b. $$\frac{1}{h}\phi\left(\frac{x-x^{(i)}}{h}\right)=\frac{1}{2\pi h}\left(\frac{\sin\left((x-x^{(i)})/(2h)\right)}{(x-x^{(i)})/(2h)}\right)^2.$$

провести численный эксперимент или статистическое имитационное моделирование и представить соответствующие графики. Провести анализ полученных результатов и представить его в виде выводов по проделанной работе.

Код программы

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
import warnings
warnings.filterwarnings('ignore')
import sys
import io

sys.stdout = io.TextIOWrapper(sys.stdout.buffer, encoding='utf-8')

def vkernel_var2(x0, XN, h_N, kl_kernel):
    p_ = np.zeros(len(x0))

    for i, x in enumerate(x0):
        u = (x - XN) / h_N

        if kl_kernel == 5:  # Окно Коши (вариант 2.а)
            # 1/(πh) * 1/(1 + ((x-x⁽ⁱ⁾)/h)²)
            kernel_vals = 1 / (np.pi * (1 + u**2)) / h_N

        elif kl_kernel == 6:  # Окно sinc² (вариант 2.б)
            # 1/(2πh) * [sin(((x-x⁽ⁱ⁾)/(2h)) / ((x-x⁽ⁱ⁾)/(2h)))]²
            mask = np.abs(u) < 1e-10
            kernel_vals = np.zeros_like(u)
            kernel_vals[mask] = 1 / (2 * np.pi * h_N)
            u_not_zero = u[~mask]
```

```
            kernel_vals[~mask] = (1 / (2 * np.pi * h_N)) * (np.sin(u_not_zero/2)
/ (u_not_zero/2))**2

        else:
            kernel_vals = norm.pdf(u) / h_N  # Гауссово по умолчанию

        p_[i] = np.mean(kernel_vals)

    return p_

#Исходные данные
n = 1  # размерность вектора наблюдений
N = 1000  # количество используемых для оценки векторов
r = 0.5
h_N = N**(-r/n)  # расчет параметра размера окна

print("Оценка плотности распределения методом Парзена")
print(f"\nПараметры: N={N}, n={n}, r={r}, h_N={h_N:.4f}")

# Генерация обучающей выборки
np.random.seed(42)
XN = -np.log(np.random.rand(N))  # Показательное распределение с параметром b=1

# Сетка и истинная плотность
x0 = np.arange(-3, 3.05, 0.05)
p_true = np.zeros(len(x0))
ind1 = x0 > 0
p_true[ind1] = np.exp(-x0[ind1])  # Показательное распределение

# Оценка плотности и вычисление MSE для обоих окон
print("\nВычисление среднеквадратичной ошибки:")

# Для окна Коши (kl_kernel=5)
p_cauchy = vkernel_var2(x0, XN, h_N, 5)
mse_cauchy = np.mean((p_true - p_cauchy)**2)
print(f"Окно Коши (kl_kernel=5): MSE = {mse_cauchy:.6e}")

# Для окна sinc² (kl_kernel=6)
p_sinc = vkernel_var2(x0, XN, h_N, 6)
mse_sinc = np.mean((p_true - p_sinc)**2)
print(f"Окно sinc² (kl_kernel=6): MSE = {mse_sinc:.6e}")

# Визуализация результатов
plt.figure(figsize=(15, 10))

# График 1: Оба метода вместе
plt.subplot(2, 2, 1)
plt.plot(x0, p_true, 'b-', linewidth=3, label='Истинная плотность')
plt.plot(x0, p_cauchy, 'r--', linewidth=2, label=f'Парзен-Коши
(MSE={mse_cauchy:.2e})')
```

```
plt.plot(x0, p_sinc, 'g--', linewidth=2, label=f'Парзен-sinc²
(MSE={mse_sinc:.2e})')
plt.xlabel('x')
plt.ylabel('Плотность вероятности')
plt.title('Сравнение методов оценки плотности\nПоказательное распределение')
plt.legend()
plt.grid(True, alpha=0.3)

# График 2: Только окно Коши
plt.subplot(2, 2, 2)
plt.plot(x0, p_true, 'b-', linewidth=3, label='Истинная плотность')
plt.plot(x0, p_cauchy, 'r--', linewidth=2, label='Оценка Парзена (Коши)')
plt.xlabel('x')
plt.ylabel('Плотность вероятности')
plt.title('Оценка плотности: окно Коши\n' + r'$\frac{1}{h}\phi(u) = \frac{1}{\pi
h} \cdot \frac{1}{1 + u^2}$')
plt.legend()
plt.grid(True, alpha=0.3)

# График 3: Только окно sinc²
plt.subplot(2, 2, 3)
plt.plot(x0, p_true, 'b-', linewidth=3, label='Истинная плотность')
plt.plot(x0, p_sinc, 'g--', linewidth=2, label='Оценка Парзена (sinc²)')
plt.xlabel('x')
plt.ylabel('Плотность вероятности')
plt.title('Оценка плотности: окно sinc²\n' + r'$\frac{1}{h}\phi(u) =
\frac{1}{2\pi h} \left[\frac{\sin(u/2)}{u/2}\right]^2$')
plt.legend()
plt.grid(True, alpha=0.3)

# График 4: Гистограмма данных
plt.subplot(2, 2, 4)
plt.hist(XN, bins=40, density=True, alpha=0.5, color='gray', label='Гистограмма
данных')
plt.plot(x0, p_true, 'b-', linewidth=2, label='Истинная плотность')
plt.xlabel('x')
plt.ylabel('Плотность вероятности')
plt.title('Гистограмма данных и истинная плотность')
plt.legend()
plt.grid(True, alpha=0.3)

plt.tight_layout()
plt.show()

# Анализ оптимальности параметра h
print("\nАнализ зависимости от параметра h:")
h_range = np.logspace(-2, 0, 30)  # h от 0.01 до 1
mse_cauchy_range = []
mse_sinc_range = []

for h in h_range:
```

```
    p_cauchy_temp = vkernel_var2(x0, XN, h, 5)
    p_sinc_temp = vkernel_var2(x0, XN, h, 6)
    mse_cauchy_range.append(np.mean((p_true - p_cauchy_temp)**2))
    mse_sinc_range.append(np.mean((p_true - p_sinc_temp)**2))

# Найдем оптимальные h
opt_idx_cauchy = np.argmin(mse_cauchy_range)
opt_idx_sinc = np.argmin(mse_sinc_range)
h_opt_cauchy = h_range[opt_idx_cauchy]
h_opt_sinc = h_range[opt_idx_sinc]

print(f"Оптимальное h для окна Коши: {h_opt_cauchy:.4f}, MSE =
{mse_cauchy_range[opt_idx_cauchy]:.6e}")
print(f"Оптимальное h для окна sinc²: {h_opt_sinc:.4f}, MSE =
{mse_sinc_range[opt_idx_sinc]:.6e}")
print(f"Рекомендуемое h_N = N^(-r/n) = {h_N:.4f}")

# График зависимости MSE от h
plt.figure(figsize=(12, 5))
plt.semilogx(h_range, mse_cauchy_range, 'r-', linewidth=2, label='Окно Коши')
plt.semilogx(h_range, mse_sinc_range, 'g-', linewidth=2, label='Окно sinc²')
plt.axvline(h_opt_cauchy, color='r', linestyle='--', alpha=0.7, label=f'h_opt
Коши = {h_opt_cauchy:.3f}')
plt.axvline(h_opt_sinc, color='g', linestyle='--', alpha=0.7, label=f'h_opt sinc²
= {h_opt_sinc:.3f}')
plt.axvline(h_N, color='b', linestyle=':', alpha=0.7, label=f'h_N = {h_N:.3f}')
plt.xlabel('Параметр сглаживания h')
plt.ylabel('Среднеквадратичная ошибка (MSE)')
plt.title('Зависимость MSE от параметра h')
plt.legend()
plt.grid(True, alpha=0.3)
plt.show()
```

Используемая формула

$$f_n(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}I\left(\frac{x - x_i}{h}\right)$$

$$MSE = \frac{1}{M}\sum_{i}(\hat{p}(x_i) - p(x_i))^2$$

Результаты выполнения задания

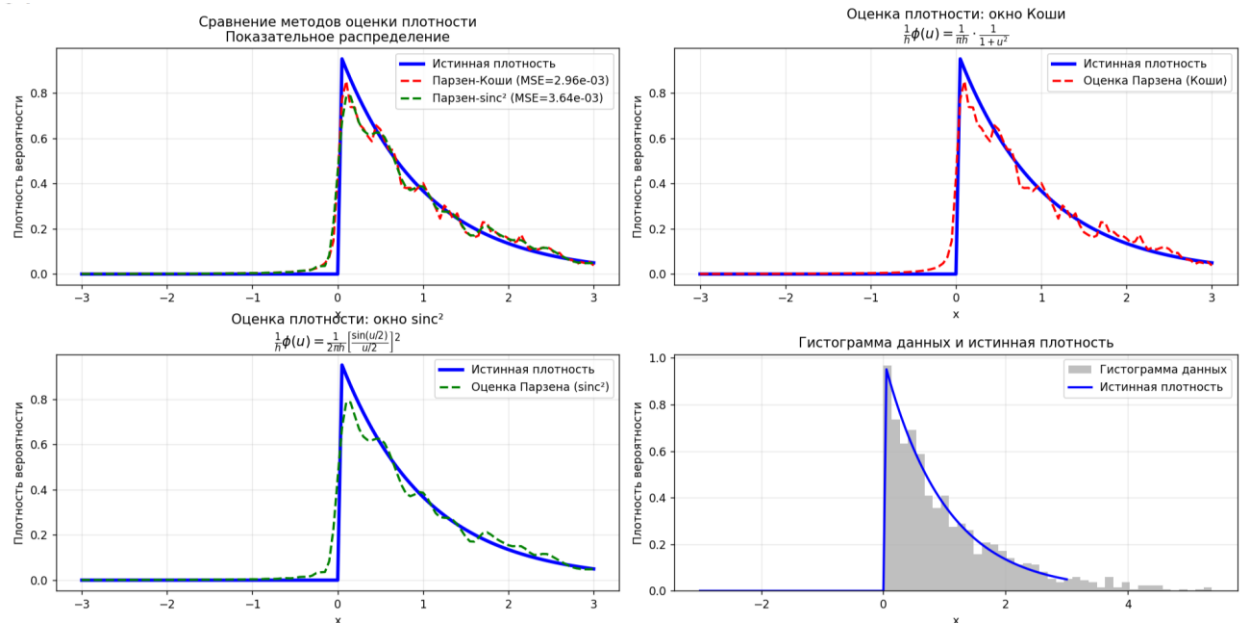

Рисунок 1.

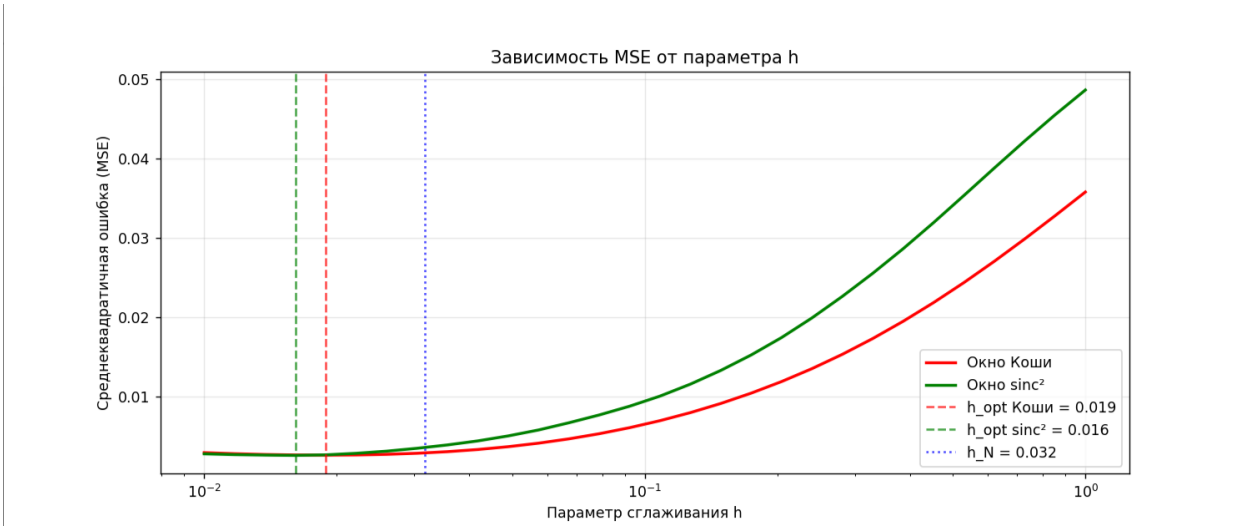

Рисунок 2.

Ответы на контрольные вопросы

1. Минимум среднеквадратичной ошибки достигается при следующих значениях параметра h:
   - Для окна Коши: h = 0.0189 (MSE = 2.679248e-03)
   - Для окна $sinc^2$: h = 0.0161 (MSE = 2.648047e-03)
2. Окно $sinc^2$ обеспечивает оптимальную оценку плотности распределения.

Выводы

В ходе лабораторной работы был исследован процесс оценки плотности распределения одномерной случайной величины методом Парзена с использованием двух типов оконных функций. Установлено, что при малых

значениях параметра сглаживания h оценка плотности получается неустойчивой из-за высокой дисперсии, а при больших – чрезмерно сглаженной вследствие роста смещения.

При оптимальных значениях параметра h (0.0189 для окна Коши и 0.0161 для окна $sinc^2$) достигается наилучшее совпадение оцененной и истинной плотностей, что подтверждается графиками сравнения, где сохраняется характер показательного распределения. Окно $sinc^2$ продемонстрировало лучшие результаты с минимальной среднеквадратичной ошибкой (2.648047e-03) по сравнению с окном Коши (2.679248e-03).

Использованные оконные функции Парзена (Коши и $sinc^2$) показали эффективность для непараметрического оценивания плотности распределения, при этом оптимальные значения параметра сглаживания оказались меньше теоретически рекомендуемого $h_N = 0.0316$, что указывает на необходимость индивидуального подбора данного параметра для конкретного распределения.