

Модели дискретных сигналов. Задача 2 (отборочный этап)

Постановка задачи

Известно, что

$$x_k = A \cos(\omega k), \quad \omega \in [0, \pi],$$

и выполнена периодичность

$$x_{k+N} = x_k \quad \forall k \in \mathbb{Z}.$$

В файле `B.txt` дана зашумлённая последовательность $\{y_k\}_{k=0}^{N-1}$:

$$y_k = x_k + \varepsilon_k,$$

где ε_k — равномерно распределённый шум.

Требуется:

1. определить ω ;
2. найти коэффициент A , минимизирующий среднеквадратичную ошибку

$$E(A) = \frac{1}{N} \sum_{k=0}^{N-1} (y_k - x_k)^2 = \frac{1}{N} \sum_{k=0}^{N-1} (y_k - A \cos(\omega k))^2.$$

Численные ответы представить с точностью до 0.0001 и приложить рассуждения, листинг и графики.

1 Использование условия периодичности

Так как $x_{k+N} = x_k$ при всех k , то

$$\cos(\omega(k + N)) = \cos(\omega k) \quad \forall k.$$

Достаточно потребовать периодичность аргумента косинуса по 2π :

$$\omega N = 2\pi m, \quad m \in \mathbb{Z},$$

откуда

$$\omega = \frac{2\pi m}{N}. \tag{1}$$

С учётом $\omega \in [0, \pi]$ получаем ограничение на m :

$$0 \leq \frac{2\pi m}{N} \leq \pi \quad \Rightarrow \quad 0 \leq m \leq \frac{N}{2}.$$

Значит, ω выбирается из конечного набора кандидатов (1).

2 Оптимальный A при фиксированном ω

При фиксированном ω функция ошибки равна

$$E(A) = \frac{1}{N} \sum_{k=0}^{N-1} (y_k - Ac_k)^2, \quad c_k = \cos(\omega k).$$

Это квадратичная функция по A . Найдём минимум:

$$\begin{aligned} \frac{dE}{dA} &= \frac{2}{N} \sum_{k=0}^{N-1} (y_k - Ac_k)(-c_k) = 0 \\ \Rightarrow \quad \sum_{k=0}^{N-1} y_k c_k &= A \sum_{k=0}^{N-1} c_k^2, \end{aligned}$$

откуда

$$A^*(\omega) = \frac{\sum_{k=0}^{N-1} y_k \cos(\omega k)}{\sum_{k=0}^{N-1} \cos^2(\omega k)}. \quad (2)$$

Далее для каждого допустимого m вычисляются $\omega = \frac{2\pi m}{N}$, затем $A^*(\omega)$ по (2) и значение $E(A^*)$. Выбирается ω , дающее минимальную ошибку.

3 Численные результаты

По данным файла получено $N = 500$. Перебор $m = 0, 1, \dots, 250$ дал минимум ошибки при

$$m = 195, \quad \omega = \frac{2\pi \cdot 195}{500} = \frac{39\pi}{50} = 2.4504422698\dots$$

Оптимальный коэффициент

$$A = 40.7569637721\dots$$

Ответ (точность 0.0001)

$$[\omega = 2.4504], \quad [A = 40.7570].$$

4 Графики

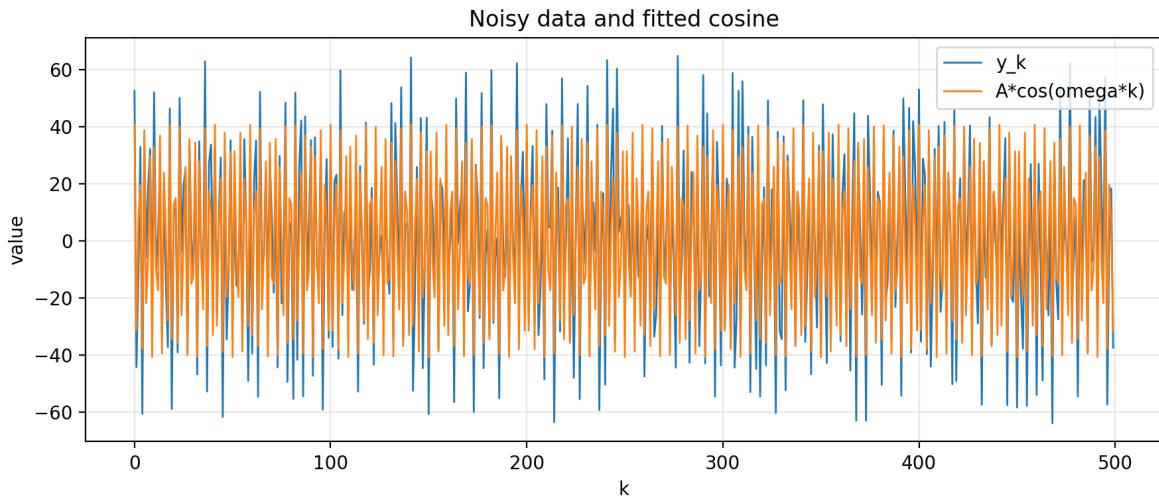


Рис. 1: Сравнение зашумлённых данных y_k и аппроксимации $A \cos(\omega k)$ при найденных ω и A .

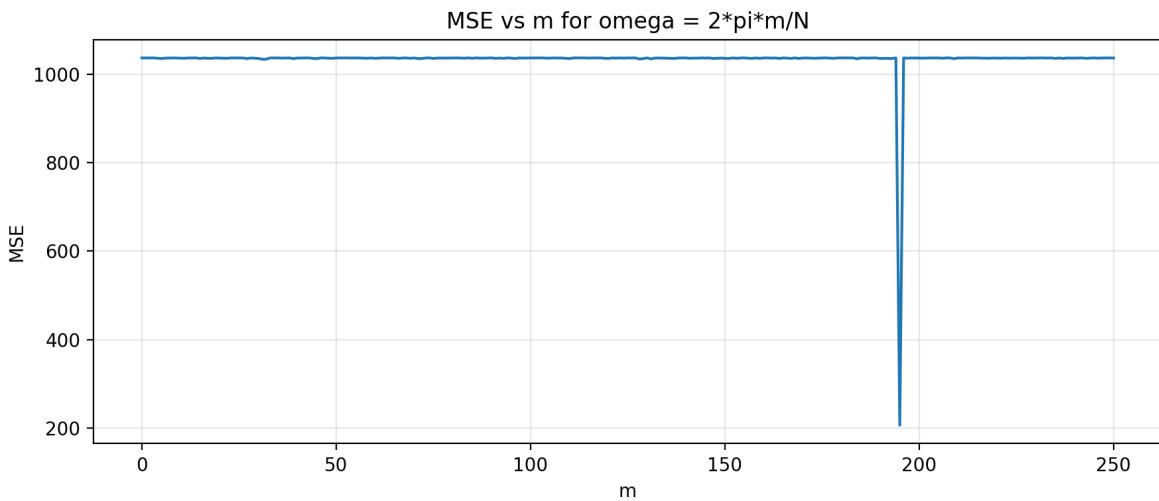


Рис. 2: Зависимость MSE от m для кандидатов $\omega = \frac{2\pi m}{N}$. Минимум достигается при $m = 195$.

5 Листинг программы (Python)

```
1 # -*- coding: utf-8 -*-
2 import numpy as np
3 import math
4 import pathlib
5 import matplotlib.pyplot as plt
6
7 path = "B.txt"
8
9 y = np.array([float(v) for v in pathlib.Path(path).read_text().split()])
```

```

10 N = len(y)
11 k = np.arange(N)
12
13 best = None
14
15 for m in range(0, N // 2 + 1):
16     omega = 2.0 * math.pi * m / N
17     c = np.cos(omega * k)
18
19     denom = float(np.dot(c, c))
20     if denom < 1e-12:
21         continue
22
23     A = float(np.dot(y, c) / denom)
24     xhat = A * c
25
26     mse = float(np.mean((y - xhat) ** 2))
27     cand = (mse, m, omega, A, xhat)
28     if best is None or cand[0] < best[0]:
29         best = cand
30
31 mse, m, omega, A, xhat = best
32
33 print("N =", N)
34 print("m =", m)
35 print("omega =", omega)
36 print("A =", A)
37 print("MSE =", mse)
38
39 print("omega_4dp =", "{:.4f}".format(omega))
40 print("A_4dp =", "{:.4f}".format(A))
41
42 plt.figure(figsize=(9, 4))
43 plt.plot(k, y, label="y_k", linewidth=1.0)
44 plt.plot(k, xhat, label="A*cos(omega*k)", linewidth=1.0)
45 plt.xlabel("k")
46 plt.ylabel("value")
47 plt.title("Noisy data and fitted cosine")
48 plt.grid(True, alpha=0.3)
49 plt.legend()
50 plt.tight_layout()
51 plt.savefig("task2_fit_plot.png", dpi=200)
52 plt.close()
53
54 ms = np.arange(0, N // 2 + 1)
55 mses = np.zeros_like(ms, dtype=float)
56
57 for idx, mm in enumerate(ms):
58     ww = 2.0 * math.pi * mm / N
59     cc = np.cos(ww * k)
60     denom2 = float(np.dot(cc, cc))
61     if denom2 < 1e-12:
62         mses[idx] = np.nan

```

```
63         continue
64     AA = float(np.dot(y, cc) / denom2)
65     mses[idx] = float(np.mean((y - AA * cc) ** 2))
66
67 plt.figure(figsize=(9, 4))
68 plt.plot(ms, mses)
69 plt.xlabel("m")
70 plt.ylabel("MSE")
71 plt.title("MSE vs m for omega = 2*pi*m/N")
72 plt.grid(True, alpha=0.3)
73 plt.tight_layout()
74 plt.savefig("task2_mse_plot.png", dpi=200)
75 plt.close()
```