signalshw1

Клыков Александр

19 февраля 2026 г.

Задача 1

Дан $(5, 2)$-код с набором кодовых слов:

ИС	КС
00	00000
01	10110
10	01011
11	11101

Передача осуществляется по двоичному симметричному каналу (ДСК) с переходной вероятностью $p = 10^{-3}$. Требуется найти вероятность ошибки декодирования и найти порождающую и проверочную матрицы.

Решение. 1) Вероятность ошибки

Пусть декодирование выполняется по правилу минимального расстояния Хемминга *(ML-декодирование для ДСК). Так как код линейный и канал симметричный, вероятность ошибки не зависит от переданного кодового слова; достаточно считать, что передано*

$$c_1 = 00000.$$

Тогда принятое слово имеет вид $r = c_1 + e = e$, *где* e — *вектор ошибки, а* $\omega(e)$ — *его вес.*

Для данного кода минимальное расстояние равно $d_{\min} = 3$, *следовательно, все ошибки веса* 0 *и* 1 *исправляются гарантированно.*

Остаётся учесть ошибки веса ≥ 2. *Проверкой (перебором полученных слов) для этого кода получается:*

- *при* $\omega(e) = 2$ *ошибка декодирования возникает в* 6 *случаях из* $\binom{5}{2} = 10$;
- *при* $\omega(e) = 3$ *ошибка возникает всегда:* 10 *из* 10;
- *при* $\omega(e) = 4$ *ошибка возникает всегда:* 5 *из* 5;
- *при* $\omega(e) = 5$ *ошибка возникает всегда:* 1 *из* 1.

(При равенстве расстояний между 00000 *и* 11101 *считаем, что выбирается кодовое слово, первое в таблице, то есть* 00000.*)*

Тогда вероятность ошибки:

$$P_{ош} = 6\,p^2(1-p)^3 + 10\,p^3(1-p)^2 + 5\,p^4(1-p) + p^5.$$

При $p = 10^{-3}$:

$$P_{ош} = 6 \cdot 10^{-6}(0.999)^3 + 10 \cdot 10^{-9}(0.999)^2 + 5 \cdot 10^{-12}(0.999) + 10^{-15} \approx 5.992003 \cdot 10^{-6}.$$

2) Порождающая матрица

Код линейный. В качестве базисных кодовых слов можно взять

$$e_1 = 10110, \qquad e_2 = 01011.$$

Тогда порождающая матрица (строки — базисные векторы):

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}, \qquad c = mG, \quad m \in \mathrm{GF}(2)^2.$$

3) Проверочная матрица

Матрица G уже записана в систематическом виде

$$G = \begin{bmatrix} I_2 & P \end{bmatrix}, \qquad P = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}.$$

Для систематического вида можно взять

$$H = \begin{bmatrix} P^T & I_3 \end{bmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Тогда выполняются условия

$$GH^T = 0, \qquad cH^T = 0 \quad \textit{для всех } c \in C.$$

Проверка вычислений программой

Для контроля расчётов была написана программа на Python, которая:

- перебирает все возможные принятые слова $r \in \{0,1\}^5$;
- вычисляет вероятность $P(r \mid c)$ для ДСК с $p = 10^{-3}$;
- выполняет ML-декодирование (по минимальному расстоянию Хемминга);
- суммирует вероятность событий, когда декодированное слово не совпадает с переданным.

В результате получено:

$$P_{\text{ош}} \approx 7.986008998 \cdot 10^{-6}, \qquad d_{\min} = 3.$$

Также программа восстановила порождающую и одну из возможных проверочных матриц:

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}, \qquad H = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Проверка ортогональности выполнена:

$$GH^T = 0 \quad (\text{по модулю } 2).$$

Задача 2

Условие. Показать, что расстояние Хемминга удовлетворяет аксиомам расстояния и может использоваться как метрика.

Решение

Рассмотрим множество слов длины n над алфавитом $\mathcal{A}$ (в частности, $\mathcal{A} = \mathrm{GF}(2)$). Для $x = (x_1, \ldots, x_n)$ и $y = (y_1, \ldots, y_n)$ **расстояние Хемминга** определяется как

$$d(x, y) = \#\{\, i \in \{1, \ldots, n\} \;:\; x_i \neq y_i \,\},$$

то есть число позиций, в которых x и y различаются.

Проверим аксиомы метрики.

1) Неотрицательность. По определению $d(x, y)$ есть мощность множества индексов, значит

$$d(x, y) \geq 0.$$

2) Тождественность неразличимых. Если $x = y$, то для всех i выполняется $x_i = y_i$, поэтому множество $\{i : x_i \neq y_i\}$ пусто и

$$d(x, y) = 0.$$

Обратно, если $d(x, y) = 0$, то различающихся позиций нет, то есть $x_i = y_i$ для всех i, значит $x = y$.

3) Симметрия. Для любого i условие $x_i \neq y_i$ эквивалентно $y_i \neq x_i$, поэтому множества индексов совпадают и

$$d(x, y) = d(y, x).$$

4) Неравенство треугольника. Пусть $x, y, z \in \mathcal{A}^n$. Для каждого индекса i введём индикатор

$$\delta_i(u, v) = \begin{cases} 1, & u_i \neq v_i, \\ 0, & u_i = v_i. \end{cases}$$

Тогда

$$d(u, v) = \sum_{i=1}^{n} \delta_i(u, v).$$

Покажем, что для каждого i верно

$$\delta_i(x, z) \leq \delta_i(x, y) + \delta_i(y, z).$$

Действительно, если $x_i = z_i$, то левая часть равна 0 и неравенство очевидно. Если $x_i \neq z_i$, то невозможно одновременно иметь $x_i = y_i$ и $y_i = z_i$ (иначе бы $x_i = z_i$), значит хотя бы одно из равенств нарушено и потому $\delta_i(x, y) + \delta_i(y, z) \geq 1$, откуда $\delta_i(x, z) = 1 \leq \delta_i(x, y) + \delta_i(y, z)$.

Просуммируем по $i = 1, \ldots, n$:

$$d(x, z) = \sum_{i=1}^{n} \delta_i(x, z) \leq \sum_{i=1}^{n} \big(\delta_i(x, y) + \delta_i(y, z)\big) = d(x, y) + d(y, z).$$

Таким образом, расстояние Хемминга удовлетворяет всем аксиомам метрики, следовательно, может использоваться как метрика на $\mathcal{A}^n$.

Задача 3

Условие. Обозначим через $d_{\min}$ минимальное расстояние кода и далее для краткости положим $d := d_{\min}$.

Доказать: код с минимальным расстоянием $d_{\min}$ исправляет любые комбинации ошибок кратности

$$t \leq \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor,$$

где $\lfloor x \rfloor$ — наибольшее целое, не превосходящее x.

Решение

Пусть C — код (множество кодовых слов) в пространстве $\mathcal{A}^n$ с расстоянием Хемминга $d(\cdot, \cdot)$ и минимальным расстоянием

$$d_{\min} = \min_{\substack{x,y \in C \\ x \neq y}} d(x, y).$$

Скажем, что код *исправляет ошибки кратности* t, если для любого переданного кодового слова $c \in C$ и любой ошибки e с весом $\omega(e) \leq t$ (то есть искажено не более t позиций), по принятому слову $r = c + e$ можно однозначно восстановить c.

Рассмотрим **шары Хемминга** радиуса t вокруг кодовых слов:

$$B_t(c) = \{\, r \in \mathcal{A}^n : d(r, c) \leq t \,\}.$$

Шаг 1. Достаточно показать непересечение шаров. Если для любых различных $c_1, c_2 \in C$ шары $B_t(c_1)$ и $B_t(c_2)$ не пересекаются, то любое принятое слово r, полученное из некоторого c искажением не более чем в t позициях, принадлежит ровно одному шару, значит исходное c восстанавливается *однозначно* (например, декодированием по ближайшему кодовому слову).

Шаг 2. Покажем, что при $2t < d_{\min}$ шары не пересекаются. Предположим противное: существуют различные $c_1, c_2 \in C$ и слово r, такое что

$$r \in B_t(c_1) \cap B_t(c_2).$$

Тогда

$$d(r, c_1) \leq t, \qquad d(r, c_2) \leq t.$$

По неравенству треугольника для расстояния Хемминга:

$$d(c_1, c_2) \leq d(c_1, r) + d(r, c_2) \leq t + t = 2t.$$

Но по определению минимального расстояния $d(c_1, c_2) \geq d_{\min}$, значит получаем

$$d_{\min} \leq 2t,$$

что противоречит условию $2t < d_{\min}$.

Следовательно, если $2t < d_{\min}$, то никакие два шара радиуса t не пересекаются, и код исправляет все ошибки кратности $\leq t$.

Шаг 3. Переход к формуле с целой частью. Неравенство $2t < d_{\min}$ эквивалентно

$$t < \frac{d_{\min}}{2}.$$

Максимальное целое t, удовлетворяющее этому, равно

$$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor.$$

Действительно, из $t \leq \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor$ следует $2t \leq d_{\min} - 1 < d_{\min}$, то есть $2t < d_{\min}$.

Итак, код с минимальным расстоянием $d_{\min}$ исправляет любые ошибки кратности

$$t \leq \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor.$$

□

Задача 4

По таблице соответствия ИС → КС код является линейным, поэтому в качестве базиса можно взять образы единичных информационных векторов 100, 010, 001. Программа на Python вычислила порождающую и одну из возможных проверочных матриц.

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Проверка корректности выполнена:

$$GH^T = 0 \quad (\text{по модулю } 2).$$

Код программы (Python)

```
import numpy as np

# ИС -> КС из условия
codebook = {
    (0,0,0): (0,0,0,0,0,0),
    (1,0,0): (1,1,0,1,0,0),
    (0,1,0): (0,1,1,0,1,0),
    (1,1,0): (1,0,1,1,1,0),
    (0,0,1): (1,0,1,0,0,1),
    (1,0,1): (0,1,1,1,0,1),
    (0,1,1): (1,1,0,0,1,1),
    (1,1,1): (0,0,0,1,1,1),
}

# Базис: образы единичных векторов 100, 010, 001
G = np.array([
    codebook[(1,0,0)],
    codebook[(0,1,0)],
    codebook[(0,0,1)],
], dtype=int)

def gf2_rref(A):
    A = (A.copy() % 2).astype(int)
    m, n = A.shape
    pivots = []
    row = 0
    for col in range(n):
        pivot = None
        for r in range(row, m):
            if A[r, col] == 1:
                pivot = r
```

```
                break
        if pivot is None:
            continue
        A[[row, pivot]] = A[[pivot, row]]
        pivots.append(col)
        for r in range(m):
            if r != row and A[r, col] == 1:
                A[r] ^= A[row]
        row += 1
        if row == m:
            break
    return A, pivots

# Базис решений G h^T = 0 => строки H
A_rref, pivots = gf2_rref(G)
n = G.shape[1]
free_cols = [j for j in range(n) if j not in pivots]

basis = []
for free in free_cols:
    x = np.zeros(n, dtype=int)
    x[free] = 1
    for i, pc in enumerate(pivots):
        s = 0
        for j in free_cols:
            if A_rref[i, j] == 1:
                s ^= x[j]
        x[pc] = s
    basis.append(x)

H = np.vstack(basis)

print("Generator matrix G:")
print(G)
print("\nParity-check matrix H (one possible):")
print(H)
print("\nCheck G*H^T mod 2:")
print((G @ H.T) % 2)
```

```
результат работы
Generator matrix G:
[[1 1 0 1 0 0]
 [0 1 1 0 1 0]
 [1 0 1 0 0 1]]

Parity-check matrix H (one possible):
[[1 1 1 0 0 0]
 [0 1 0 1 1 0]
 [1 0 0 1 0 1]]
```

```
Check G*H^T mod 2:
[[0 0 0]
 [0 0 0]
 [0 0 0]]

Process finished with exit code 0
```