Домашнее задание 4

Кокорин Илья, M3439

Вариант № 64

1 Задание 3

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

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

$k = 6, n = 10, r = 4, d = 3$

Определим диапазон оптимальных значений d для данного кода.

1.1 Граница Хэмминга

Граница Хемминга имеет вид $M \leq \frac{q^n}{\sum\limits_{i=0}^{t} \binom{n}{i} \cdot (q-1)^i}$, в случае двоичного кода, у которого $M = 2^k, q = 2$, принимает вид $2^k \leq \frac{2^n}{\sum\limits_{i=0}^{t} \binom{n}{i} \cdot (2-1)^i} \Rightarrow 2^k \leq \frac{2^n}{\sum\limits_{i=0}^{t} \binom{n}{i}} \Rightarrow \sum\limits_{i=0}^{t} \binom{n}{i} \leq 2^{n-k}$

При заданных $n, k = 10, 6$ принимает вид: $\sum\limits_{i=0}^{t} \binom{10}{i} \leq 16$

Побрерём перебором такое максимальное d, что при $t = \lfloor \frac{d-1}{2} \rfloor$ формула выполняется.

```
import math


def c(n, k):
        return math.factorial(n) //
                (math.factorial(k) * math.factorial(n - k))


def check_hamming(n, k, d):
        t = (d - 1) // 2
        total_sum = 0
        for i in range(t + 1):
                total_sum += c(n, i)
        return total_sum <= 2 ** (n - k)


cur_d = 1
n = 10
k = 6
```

```
while True:
        if not check_hamming(n, k, cur_d):
                print(cur_d - 1)   # 4
                break
        else:
                cur_d += 1
```

По теореме Хемминга, верхней границей для d является 4, то есть ни для какого $d' > 4$ не существует двоичный линейный (n, k)-код с минимальным расстоянием d'.

1.2 Граница Варшамова-Гилберта

Граница Варшамова-Гилберта имеет вид $q^{n-k} > \sum\limits_{i=0}^{d-2} \binom{n-1}{i} \cdot (q-1)^i$

Что в нашем случае, превращается в $2^{10-6} > \sum\limits_{i=0}^{d-2} \binom{10-1}{i} \cdot (2-1)^i \Rightarrow 16 > \sum\limits_{i=0}^{d-2} \binom{9}{i} \Rightarrow \sum\limits_{i=0}^{d-2} \binom{9}{i} < 16$

Найдём с помощью перебора такое максимальное d, для которого эта формула выполяется. Тогда, по теореме Варшамова-Гилберта, для такого d будет существовать двоичный линейный (n, k)-код с минимальным расстоянием d.

Приведём программу для поиска такой границы.

```
import math


def c(n, k):
        return math.factorial(n) //
                (math.factorial(k) * math.factorial(n - k))


def check_hilbert(n, k, d):
        total_sum = 0
        for i in range(d - 1):
                total_sum += c(n - 1, i)
        return total_sum < 2 ** (n - k)


n = 10
k = 6
cur_d = 1
while True:
        if not check_hilbert(n, k, cur_d):
                print(cur_d - 1)   # 3
                break
        else:
                cur_d += 1
```

То есть для $d = 3$ существует двоичный линейный (n, k)-код с минимальным расстоянием d.

1.3 Проверка кода на оптимальность

Тогда минимальное расстояние оптимального кода следует искать во множестве $\{d : 3 \leq d \leq 4\}$

Минимальное расстояние исследуемого кода попадает в данный диапазон.

Как следует из материалов сайта $http://www.codetables.de/$ и таблицы 3.2, лучший код для $n = 10, k = 6$ имеет $d = 3$, то есть приведённый код является оптимальным.

2 Задача 3.3

Зафиксируем $n, d = 31, 7$

2.1 Граница Хемминга

$t = \lfloor \frac{d-1}{2} \rfloor = 3$

Граница Хемминга имеет вид $M \leq \frac{q^n}{\sum\limits_{i=0}^{t} \binom{n}{i} \cdot (q-1)^i}$, в случае двоичного кода, у которого $M = 2^k, q = 2$, принимает вид $2^k \leq \frac{2^n}{\sum\limits_{i=0}^{t} \binom{n}{i} \cdot (2-1)^i} \Rightarrow 2^k \leq \frac{2^n}{\sum\limits_{i=0}^{t} \binom{n}{i}} = \frac{2^{31}}{\binom{31}{0}+\binom{31}{1}+\binom{31}{2}+\binom{31}{3}} = \frac{16777216}{39}$

Тогда $k = \lfloor log_2(\frac{16777216}{39}) \rfloor = 18$

$k = 18$ является верхней границей для k, то есть ни для какого $k' > 18$ не существует двоичного линейного (n, k)-кода с заданными параметрами.

2.2 Граница Варшамова-Гилберта

Граница Варшамова-Гилберта имеет вид $q^{n-k} > \sum\limits_{i=0}^{d-2} \binom{n-1}{i} \cdot (q-1)^i$

Что в нашем случае, превращается в $2^{31-k} > \sum\limits_{i=0}^{7-2} \binom{31-1}{i} \cdot (2-1)^i \Rightarrow 2^{31-k} > \sum\limits_{i=0}^{5} \binom{30}{i} \Rightarrow 2^{31-k} > 174437 \Rightarrow 31 - k > log_2(174437) \Rightarrow 31 - k \geq 18 \Rightarrow k \leq 13$

Тогда при $k \leq 13$ точно существует код с вышеуказанными параметрами. Значит, такой код гарантированно существует для $k = 13$. Тогда $k = 13$ является нижней границей для оптимального k.

2.3 Определение оптимального кода

Так как код с вышеуказанными параметрами гарантированно существует при $k \leq 13$ (а значит, и при $k = 13$), и точно не существует при $k > 18$, то k для оптимального кода нужно искать во множестве $\{k : 13 \leq k \leq 18\}$

Из таблицы 3.2 следует, что при $n = 31$ минимальное расстояние $d = 7$ достигается при $k = 17$, $17 \in \{k : 13 \leq k \leq 18\}$

3 Задача 3.4

Выберем $n = 25, k = 17$

3.1 Граница Хемминга

Граница Хемминга имеет вид $M \leq \frac{q^n}{\sum\limits_{i=0}^{t} \binom{n}{i} \cdot (q-1)^i}$, в случае двоичного кода, у которого $M = 2^k, q = 2$, принимает вид $2^k \leq \frac{2^n}{\sum\limits_{i=0}^{t} \binom{n}{i} \cdot (2-1)^i} \Rightarrow 2^k \leq \frac{2^n}{\sum\limits_{i=0}^{t} \binom{n}{i}} \Rightarrow \sum\limits_{i=0}^{t} \binom{n}{i} \leq 2^{n-k}$

Поберём перебором такое максимальное d, что при $t = \lfloor \frac{d-1}{2} \rfloor$ формула выполняется.

```
import math


def c(n, k):
        return math.factorial(n) //
                (math.factorial(k) * math.factorial(n - k))


def check_hamming(n, k, d):
        t = (d - 1) // 2
        total_sum = 0
```

```
        for i in range(t + 1):
                total_sum += c(n, i)
        return total_sum <= 2 ** (n - k)


cur_d = 1
n = 25
k = 17

while True:
        if not check_hamming(n, k, cur_d):
                print(cur_d - 1)  # 4
                break
        else:
                cur_d += 1
```

По теореме Хемминга, верхней границей для d является 4, то есть ни для какого $d' > 4$ не существует двоичный линейный (n, k)-код с минимальным расстоянием d'.

3.2 Граница Варшамова-Гилберта

Граница Варшамова-Гилберта имеет вид $q^{n-k} > \sum\limits_{i=0}^{d-2} \binom{n-1}{i} \cdot (q-1)^i$

Что в нашем случае, превращается в $2^{25-17} > \sum\limits_{i=0}^{d-2} \binom{25-1}{i} \cdot (2-1)^i \Rightarrow 256 > \sum\limits_{i=0}^{d-2} \binom{24}{i} \Rightarrow \sum\limits_{i=0}^{d-2} \binom{24}{i} < 256$

Найдём с помощью перебора такое максимальное d, для которого эта формула выполяется. Тогда, по теореме Варшамова-Гилберта, для такого d будет существовать двоичный линейный (n, k)-код с минимальным расстоянием d.

Приведём программу для поиска такой границы.

```
import math


def c(n, k):
 return math.factorial(n) //
                (math.factorial(k) * math.factorial(n - k))


def check_hilbert(n, k, d):
        total_sum = 0
        for i in range(d - 1):
                total_sum += c(n - 1, i)
        return total_sum < 2 ** (n - k)

n = 25
k = 17
cur_d = 1
while True:
        if not check_hilbert(n, k, cur_d):
                print(cur_d - 1)  # 3
                break
        else:
                cur_d += 1
```

То есть для $d = 3$ существует двоичный линейный (n, k)-код с минимальным расстоянием d.

3.3 Проверка кода на оптимальность

Тогда минимальное расстояние оптимального кода следует искать во множестве $\{d : 3 \leq d \leq 4\}$
Как следует из таблицы 3.2, лучший код для $n = 25, k = 17$ имеет $d = 4$, $4 \in \{d : 3 \leq d \leq 4\}$