Дмитрий Герасимов (4538)
Теория кодирования

Задание 3

8 декабря 2013 г.

Задача 1

Граница Хэмминга:

$$|\mathcal{C}| \leq \frac{q^n}{\sum\limits_{i=0}^{t} \binom{n}{i}(q-1)^i}$$

Для кода Голея: $n = 23, k = 12, d = 7, t = 3$. Подставим:

$$2^{12} \leq \frac{2^{23}}{\binom{23}{0} + \binom{23}{1} + \binom{23}{2} + \binom{23}{3}} = \frac{2^{23}}{2^{11}} = 2^{12}$$

, действительно, код удовлетворяет границе Хэмминга и более того, достигает ее.

Для кода Хэмминга: $n = 2^r - 1, k = 2^r - r - 1, t = 1$. Подставим:

$$2^{2^r - r - 1} \leq \frac{2^{2^r - 1}}{\binom{2^r-1}{0} + \binom{2^r-1}{1}} = \frac{2^{2^r-1}}{1 + 2^r - 1} = 2^{2^r - r - 1}$$

, действительно, код удовлетворяет границе Хэмминга и более того, достигает ее.

Задача 2

Как известно, код дуальный к коду Хэмминга $(2^r - 1, 2^r - r - 1)$ — симплексный код, в котором $n' = n = 2^r - 1$, $k' = r$ и минимальное расстояние $d' = 2^{r-1}$.

- Воспользуемся границей Хэмминга.

  $$|\mathcal{C}| \leq \frac{2^{2^r-1}}{\sum\limits_{i=0}^{2^{r-2}} \binom{2^r-1}{i}}$$

  Для упрощения расчетов занизим оценку, то есть оценим сверху знаменатель. Воспользуемся известной верхней границей:

  $$\sum_{i=0}^{\alpha n} \binom{n}{i} = 2^{nh(\alpha) - \frac{1}{2}\log_2 n + O(1)}$$

  Поставляя $n = 2^r - 1$ и $\alpha = \frac{1}{4}$ получим:

  $$2^{(2^r-1)h(\frac{1}{4}) - \frac{1}{2}\log_2(2^r-1) + O(1)} \approx 2^C \frac{2^{2^r h(\frac{1}{4})}}{\sqrt{2^r - 1}}$$

  Теперь, границу Хэмминга можно оценить снизу как

  $$\frac{\sqrt{2^r-1}}{2^C} \frac{2^{2^r-1}}{2^{2^r h(\frac{1}{4})}} \approx \frac{\sqrt{2^r}}{2^C} 2^{2^r(1 - h(\frac{1}{4}))}$$

  , то есть получили, что количество кодовых слов, возможное для кода с такими параметрами, растет как минимум как 2^{2^r}, в то время, как количество кодовых слов в симплексном коде равно всего лишь 2^r.

- Воспользуемся асимптотической границей Хэмминга. Относительное минимальное расстояние $\delta = \frac{d'}{n'} \approx \frac{1}{2}$, предельная скорость кода рассчитывается как $R_H(\frac{1}{2}) = 1 - h(\frac{1}{4})$. Однако с увеличением r, скорость $(2^r - 1, r)$-симплексного кода рассчитывается как $R = \frac{r}{2^r - 1}$, то есть убывает экспоненциально.

- Воспользуемся асимптотической границей Плоткина:

$$R_P(\delta) = 1 - 2\delta = 0$$

, что значительно точнее асимптотической границы Хэмминга для данного семейства кодов, и действительно, так как с увеличением r скорость симплексного кода убывает экспоненциально, в пределе она будет давать 0.

- Воспользуемся асимптотической границей Басалыго-Элайеса:

$$R_B(\delta) = 1 - h(\frac{1}{2} - \frac{1}{2}\sqrt{1 - 2\delta})$$

Она также при увеличении r даст предельную скорость кода 0.

Задача 3

Вариант 5: $n = 19; k = 12; d = 4$

- Применим границу Хэмминга для получения верхней границы на k:

$$2^k \leq \frac{2^{19}}{\binom{19}{0} + \binom{19}{1}} = 26214.4$$

, таким образом, получили $k \leq 14$

- Воспользуемся границей Варшамова-Гилберта:

$$2^{19-k} > \sum_{i=0}^{2} \binom{18}{i}$$

, из которого получаем $k \geq 11$

- Можно попытаться уточнить уже полученный диапазон с помощью границы Грайсмера, однако ничего нового она не дает:
  - $k = 11$: $n \geq 15$
  - $k = 12$: $n \geq 16$
  - $k = 13$: $n \geq 17$
  - $k = 14$: $n \geq 18$

Таким образом, получили $11 \leq k \leq 14$.

Задача 4

Вариант 5: $n = 19; k = 12; d = 4$

- Применим границу Хэмминга для получения верхней границы на d:

  $$2^{12} \leq \frac{2^{19}}{\sum\limits_{i=0}^{t} \binom{19}{i}}$$

  , получим, что максимальное t, при котором неравенство выполняется, равно 1, значит, $d \leq 4$.

- Воспользуемся границей Варшамова-Гилберта:

  $$2^7 > \sum_{i=0}^{d-2} \binom{18}{i}$$

  , из которого получаем $d \geq 3$

- Воспольуемся границей Грайсмера:

  $$19 \geq \sum_{i=0}^{11} \left\lceil \frac{d}{2^i} \right\rceil$$

  Получаем $d \leq 4$, то есть ничего нового.

Таким образом, $3 \leq d \leq 4$.

Задача 5

n	k	d	В-Г	Хэмминг
8	4	4	3	4
10	5	4	3	4
12	6	4	3	4
14	7	4	4	6
16	8	5	4	6
18	9	6	4	6
20	10	6	4	6
22	11	7	5	8
24	12	8	5	8
26	13	7	5	8
28	14	8	5	8
30	15	8	6	10
32	16	8	6	10
34	17	8	6	10
36	18	8	6	10
38	19	8-9	7	10
40	20	8-9	7	12

Задача 6

Зафиксируем k, тогда можно построить последовательность асимптотически хороших кодов $\{(ik, k)\}_{i=0}^{\infty}$, взяв в качестве k базисных кодовых слов слово, в котором первые i символов — единицы, а остальные $n-i$ элеменов нули, и его $k-1$ сдвиг на i позиций. К примеру, для $(8, 4)$-кода получим порождающую матрицу

$$G_{8,4} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Очевидно, данный код будет линейным кодом с минимальным расстоянием i. Для такой последовательности кодов получим константное относительное минимальное расстояние, равное $\frac{i}{ik} = \frac{1}{k}$, значит, последовательность кодов асимптотически хорошая.

Воспользуемся границей Грайсмера и оценим верхнюю границу на минимальное расстояние для последовательности кодов с фиксированным k.

$$ik \geq \sum_{j=0}^{k-1} \left\lceil \frac{d}{2^j} \right\rceil$$

, надо найти максимальное d, при котором неравенство выполнится. Для упрощения, избавимся от округления вверх в границе Грайсмера:

$$ik \geq \sum_{j=0}^{k-1} \left(\frac{d}{2^j} + 1 \right) \geq \sum_{j=0}^{k-1} \left\lceil \frac{d}{2^j} \right\rceil$$

, и найдем такие d, что они удовлетворяют первому неравенству. Так как мы рассматриваем асимптотические характеристики, возьмем достаточно большой i, тогда утверждается, что в качестве d можно взять $\frac{ik}{2}$ (можно считать, что i берется четное, так как характеристика ищется асимптотическая):

$$\sum_{j=0}^{k-1} \left(\frac{d}{2^j} + 1 \right) = k + \sum_{j=0}^{k-1} \frac{\frac{ik}{2}}{2^j} = k + ik \sum_{1}^{k} \frac{1}{2^j} = k + ik(1 - \frac{1}{2^k}) = k + ik - i\frac{k}{2^k}$$

Таким образом, получили, что когда

$$ik \geq k + ik - i\frac{k}{2^k}$$

$$i \geq 2^k$$

, граница Грайсмера дает минимальное расстояние, равное $\frac{ik}{2}$. Больше получить с разумной точки зрения нельзя в случае, когда $k \geq 2$, так как по принципу Дирихле не сможем получить линейный код, в котором вес всех ненулевых слов больше половины их длины.

В итоге, по границе Грайсмера получили относительное минимальное расстояние, равное $\frac{1}{2}$ для любого k, что значительно лучше $\frac{1}{k}$.

Задача 7

k	Известное n	Грайсмер
3	14	14
4	15	15
5	16	16
6	18	17
7	19	18
8	20	19
9	21	20
10	22	21
11	23	22
12	24	23
13	27	24
14	28	25
15	30	26
16	31	27
17	32	28
18	34	29
19	35	30
20	36	31
21	37	32
22	38	33
23	40	34

Задача 8

Был реализован следующий алгоритм:

1. Будем строить проверочную матрицу сразу в систематическом виде, поэтому первыми r столбцами возьмем столбцы, формирующие единичную матрицу.

   Этот шаг занимает $O(r^2)$ операций.

2. Далее будем поддерживать множество векторов `allowed`, из которого разрешено выбирать следующий вектор-столбец для проверочной матрицы. Изначально в этом множестве есть все векторы, не являющиеся линейными комбинациями $d-2$ и менее векторов из первых r (так как если вектор является линейной комбинацией $d-2$ других, то в матрице будет $d-1$ линейно зависимый столбец).

   Генерация всех сочетаний $\binom{r}{d-2}$ занимает время $O(r\binom{r}{d-2})$, с учетом того, что надо считать суммы векторов, получаем $O(r^3\binom{r}{d-2})$, что можно оценить сверху как $O(r^3\binom{r}{r/2})$, и воспользовавшись тем, что $\binom{2n}{n} \sim \frac{4^n}{\sqrt{\pi n}}$, получим асимптотику $O(r^{2.5}2^r)$

3. Начинаем набирать оставшиеся $n-r$ столбец в проверочную матрицу: делаем $n-r$ таких итераций: на очередной итерации берем произвольный вектор из `allowed`, обозначим его за `current` и добавим в проверочную матрицу. Далее

надо обновить множество `allowed`, для этого надо удалить из него все сочетания из не более чем $d-2$ столбцов проверочной матрицы, в которых есть `current`.

На i-й итерации данного шага в проверочной матрице находится $r+i-1$ элемент, из которых мы выбираем все сочетания размером не более $d-3$ (и дописываем к ним столбец `current`, чтобы получить сочетания размера не более $d-2$), суммируем столбцы в каждом сочетании и удаляем из `allowed`, то есть получаем $\sum\limits_{l=0}^{d-3} r^3\binom{r+i-1}{l}$ операций. Рассмотрим последнюю итерацию, на ней в матрице уже будет $n-1$ столбец, и будет выполнено $\sum\limits_{l=0}^{d-3} r^3\binom{n-1}{l}$ операций. Заметим, что $d \leq \frac{n}{2}$ при $k \geq 2$, значит, можно оценить сумму как $r^3 \sum\limits_{l=0}^{(n-1)/2} \binom{n-1}{l}$, что равно $r^3 2^{n-2}$. Оценим сверху остальные $n-r$ слагаемых последним, и получим суммарную асимптотику данного шага в $O((n-r)r^3 2^{n-2})$

В итоге, очевидно, асимптотика последнего шага доминирует над первыми двумя, а если не учитывать полиномиальный множитель, получаем сложность алгоритма в худшем случае $O^*(2^n)$.

Также заметим, что на поддержание списка `allowed` необходимо $O^*(2^r)$ памяти, что может быть критично при получении длинных кодов с малым числом кодовых слов.

Моя реализация очень быстро (менее секунды) генерирует коды, в которых $n-k$ мало, к примеру, меньше 20. Коды, в которых $n-k$ больше 25 сложно генерировать из-за ограниченной оперативной памяти.

Примеры кодов, которые можно получить за время около минуты: $(41, 21)$, $(60, 38)$, $(200, 180)$, $(80, 58)$.

В целом, учитывая ограничение $n-k \leq 25$, можно генерировать коды длинее 40 с расстояниями от 9 и меньше (например, 6 при $n = 100$).

Задача 9

Пусть у нас есть (n, k, d)-код.

1. При удалении кодового слова минимальное расстояние не может ухудшиться, поэтому в таком случае мы можем получить $(n, k-1, d)$-код.

2. При вычеркивании произвольного столбца из порождающей матрицы линейного кода минимальное расстояние может ухудшиться не более чем на 1, то есть получим $(n-1, k, d-1)$-код.

3. Комбинируя два предыдущих утверждения, получим, что также можем получить $(n-1, k-1, d-1)$-код, что означает, что при движении по диагоналям таблицы минимальное расстояние невозрастает.

Таким образом, после того как мы получили, что лучшее минимальное расстояние некоторого (n, k) кода равно d, мы можем обновить все значения в таблице

- слева, то есть релаксировать нижнюю границу на лучшее минимальное расстояние на d для k', меньших k
- сверху, то есть релаксировать нижнюю границу на лучшее минимальное расстояние на $d - i$ для $n' = n - i$
- справа, то есть релаксировать верхнюю границу на лучшее минимальное расстояние на $d + 1$ для k', больших k
- снизу, то есть релаксировать верхнюю границу на лучшее минимальное расстояние на $d + i$ для $n' = n + i$

, и повторять процедуру, пока значения в таблице меняются.

Вообще говоря не очень понял вопрос про «сколько оценок надо улучшить», так как в зависимости от конкретных уточненных расстояний получим разное количество клеток, которые будут «открыты» «автоматически».