Алгебраическая теория кодирования с применением Sage

31 октября 2023 г.

Оглавление

Глава 1

ВВЕДЕНИЕ

Основные понятия теории кодирования. Блочные коды. Основные параметры блочного кода. Метрика Хэмминга. Минимальное расстояние кода. Коды с обнаружением и исправлением ошибок, связь с минимальным расстоянием.

Рассматривается задача обнаружения и исправления ошибок при передачи сообщений по каналам связи.

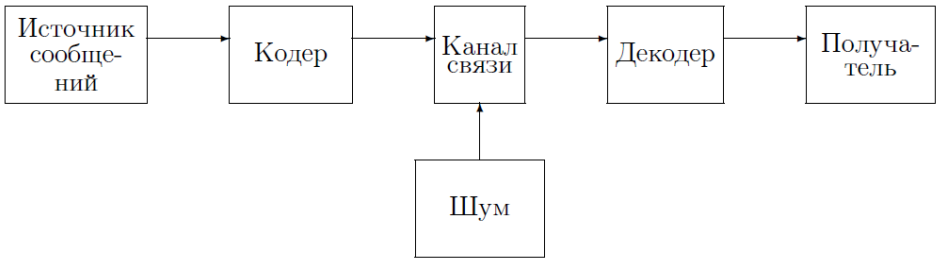

Рис. 1.1: Система связи по каналу с шумом.

Мы рассматриваем передачу сообщений блоками размера n над некоторым алфавитом X. Общая схема работы подразумевается следующей:

- Источник формирует сообщение $u = (u_1, u_2, \ldots, u_k)$ длины k.
- Кодер добавляет проверочные символы к сообщению (само сообщение тоже может изменятся).
- Сообщение передается по каналу связи, в котором возможны помехи.
- Декодер проверяет наличие ошибок, при возможности исправляет их и предает декодированное сообщение получателю.

Пример 1.1. Код с повторениями. Пользователь хочет передать сообщение 1 или 0 (информационное сообщение, $k = 1$). Для уменьшения ошибки символ дублируется 5 раз. По каналу связи будет передано либо 00000, либо 11111.

Если будет получено, например, сообщение 01000, то значит в момент передачи произошла ошибка (обнаружение ошибки).

Если вероятность ошибки в каждом символе меньше 0.5, то вероятнее всего было передано сообщение 00000 (исправление ошибки).

Пример 1.2. Код проверки на четность. Пользователь передает сообщение $(u_1, u_2, \ldots, u_k)$, состоящее из 0 и 1. Кодер добавляет один дополнительный символ $u_{k+1} \in \{0, 1\}$, так чтобы сумма $u_1 + u_2 \ldots + u_k + u_{k+1}$ была четная. Если в канале произойдет одна ошибка (или любое нечетное количество ошибок), то сумма станет нечетной и мы зафиксируем наличие ошибки.

В этом случае восстановить сообщение мы не сможем. Такие коды используют в системах, где вероятность ошибки очень маленькая, например, в вычислительных машинах для контроля передач информации между регистрами и для контроля считываемой информации в оперативной памяти.

Пример 1.3. ISBN-код. Международный стандартный книжный номер (англ. International Standard Book Number, сокращённо – англ. ISBN) – уникальный номер книжного издания, необходимый для распространения книги в торговых сетях и автоматизации работы с изданием. Пример ISBN-кода: ISBN 3-88053-002-5.

Изначально ISBN имел длину из 10 символов (сейчас 13), символы состоят из цифр от "0"до "9"и буквой "X". Между информационными символами стоят дефисы, которые нужны для удобства восприятия длинного числа, буква "X"служит для обозначения числа 10 и может стоять только на последней позиции.

Код состоит из четырех частей (между которыми располагается дефис): идентификаторы группы, издателя, книги для издателя, и контрольная цифра. Идентификатор группы используется для обозначения страны, географического региона, языка и прочее.Четвертая, заключительная часть (контрольная цифра), используется в коде алгоритме другими цифрами для получения поддающегося проверке ISBN. Количество цифр, содержащееся в первых трех частях, может быть различным, но контрольная цифра всегда содержит один символ (расположенный между "0"и "9"включительно, или "X"для величины 10), а само ISBN в целом имеет длину тринадцать символов (десять чисел плюс три дефиса, разделяющих три части ISBN).

Последний символ – проверочный. Если $x_1, x_2, \ldots, x_9$ – информационные символы, то проверочный символ x_{10} выбирается из условия

$$\sum_{i=1}^{10} i \cdot x_i = 0 \mod 11.$$

1.1 Блоковые коды. Расстояние Хемминга

В нашем курсе мы рассматриваем только блоковые коды. Это означает, что сообщения передаются блоками длины n над некоторым алфавитом F, состоящим из q элементов.

В качестве алфавита берут обычно какой-либо алгебраический объект, в приложениях это чаще всего конечное поле. Есть исследования, в которых алфавит это конечные кольца, группы и полугруппы, лупы.

Определение 1.1. Кодом длины n называется любое непустое подмножество $C \in F^n$. Если количество элементов $|C| = M$, то говорят, что C является (n, M)-кодом.

Элементы кода называются кодовыми словами.

Величина $k = \log_q M$ называется размерностью (или информационной длиной) кода C. Фактически k/n показывает долю полезных (информационных) символов, которые передаются по каналу связи.

Фактически декодер проверяет принадлежит ли принятое слово множеству C или нет. Если слово не принадлежит C, то значит произошла ошибка. Далее будем решать следующие задачи:

1. Как задать C, чтобы проверка принадлежности была простой.

2. Если произошла ошибка, то можно ли восстановить исходное сообщение.

3. Как увеличить долю информационных символов, сохраняя корректирующие возможности кода.

Определение 1.2. *Расстояние Хэмминга* на множестве F^n определяется следующим образом:

$$d(x, y) = |\{i | x_i \neq y_i\}|,$$

где $x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_n)$, $x, y \in X^n$.

Теорема 1.1. Расстояние Хэмминга задает на F^n структуру метрического пространства.

Определение 1.3. *Минимальное расстояние (кодовое расстояние)* кода определяется следующим образом:

$$d(C) = \min d(x, y),$$

где минимум берется по всем парам элементов $x, y \in C$, $x \neq y$.

Теорема 1.2. Если кодовое расстояние $d(C) = 2t+1$, то код может исправлять t и обнаруживать $2t$ ошибок.

Если кодовое расстояние $d(C) = 2t$, то код может исправлять $t - 1$ и обнаруживать $2t - 1$ ошибку.

Чтобы вычислить расстояние Хэмминга между двумя векторами, можно пройтись циклом и подсчитать количество несовпадающих позиций.

Далее мы будем рассматривать линейные коды, для которых вводится понятие веса Хэмминга. Вес Хемминга $wt(x)$ – это количество ненулевых координат вектора. С помощью Sage вес Хемминга и расстояние можно вычислить следующим образом.

In [1]:
```
# Создадим два вектора над полем GF(2)
x=vector(GF(2), [1, 0, 1, 1, 0]);
y=vector(GF(2), [0, 1, 0, 1, 1]);
# Вес Хемминга вектора x
print('Вес Хемминга вектора x=', x.hamming_weight())
# Расстояние Хемминга между векторами x и y
print('Расстояние Хемминга между x и y=',(x-y).hamming_weight())
```

Out [1]: Вес Хемминга вектора x= 3

Расстояние Хемминга между x и y= 4

Пример. Найдите все слова, которые находятся на расстоянии 3 от слова 1010.

In [2]:
```
    # Создадим вектор над полем GF(2)
    x=vector(GF(2), [1, 0, 1, 0]);
    # Найдем все слова перебором
    for k in range(2):
        for l in range(2):
            for i in range(2):
                for j in range(2):
                    y=vector(GF(2), [k, l, i, j])
                    if (x-y).hamming_weight()==3:
                        print('y=', y)
```

Out [2]: y= (0, 0, 0, 1)

y= (0, 1, 0, 0)

y= (0, 1, 1, 1)

y= (1, 1, 0, 1)

1.2 Упражнения

В качестве алфавита рассматривается множество $F = GF(2) = \{0, 1\}$.

1.1. Вычислите $d(11001, 01110)$.

1.2. Вычислите $d(0000, 0110)$.

1.3. Для данного множества $C \subset F^n$ найти (минимальное) кодовое расстояние. Для каждого из кодов найти: число ошибок, которые код обнаруживает и исправляет.

1) $C = \{101010, 010110, 000001\}$, 2) $C = \{01101010, 11000110, 00011001, 10101100\}$.

1.4. Для данного множества $C \subset F^n$ найти (минимальное) кодовое расстояние. Для каждого из кодов найти: число ошибок, которые код обнаруживает и исправляет.

1) $C = \{11000, 10101, 01110\}$, 2) $C = \{111100, 110011, 001111\}$.

1.5. Найдите все слова, которые находятся на расстоянии 3 от слова 1010 в F^4.

1.6. Найдите все слова, которые находятся на расстоянии 3 от слова 10101 в F^5.

1.7. Пусть $C = \{000000, 100110, 010101, 001011, 101101, 011110, 110011, 111000\}$. Учитывая, что это $[6, 8, 3]_2$ код, восстановите сообщения 000001.

1.8. Пусть $C = \{000000, 100110, 010101, 001011, 101101, 011110, 110011, 111000\}$. Учитывая, что это $[6, 8, 3]_2$ код, восстановите сообщения 011110.

Глава 2

ЛИНЕЙНЫЕ КОДЫ

Линейные подпространство над конечным полем как коды, исправляющие ошибки. Порождающая и проверочная матрицы линейного кода. Декодирование линейного кода. Вычисления в Sage.

Для удобства кодирования и декодирования обычно рассматриваются коды с какой-либо алгебраической структурой. Наиболее важными с практической точки зрения являются линейные коды.

Далее в качестве алфавита берется конечное поле $F = GF(q)$, состоящее из q элементов.

Определение 2.1. *Линейным кодом* C называется подпространство векторного пространства F^n над полем F.

Размерность пространства C называется размерностью кода. Если k – размерность линейного пространства C и d – минимальное расстояние кода C, то говорят, что C линейный $[n, k, d]$-код над C.

Применяют также запись $[n, k]$-код и $[n, k, d]_q$-код для указания количества элементов в поле F.

Из определения ясно, что в C ровно q^k элементов.

Определение 2.2. *Весом Хэмминга* $wt(x)$ слова (вектора) $x \in F^n$ называется число ненулевых компонент x.

Теорема 2.1. Расстояние Хэмминга для слов выражается через вес Хэмминга следующим образом:

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

Если код C линеен, то

$$d(C) = \min_{x \in C, x \neq 0} wt(x).$$

Линейный код как векторное подпространство можно задать либо через систему порождающих векторов (с помощью порождающей матрицы), либо как решение систем линейных уравнений (с помощью проверочной матрицы).

2.1 Порождающая и проверочная матрица

Мы можем задать (построить) линейное подпространство с помощью порождающей матрицы

$$C = \{uG \mid u = (u_1, u_2, \ldots, u_k),\ G - k \times n \text{ матрица}\}.$$

Строки порождающей матрицы образуют базис подпространства C.

Пример.

Код проверки на четность

$$\{(u_1, u_2, \ldots, u_k, u_1 + u_2 + \ldots + u_k) = (u_1, u_2, \ldots, u_k)(E_k|A)\}.$$

$$(E_k|A) = \begin{pmatrix} 1 & 0 & \ldots & 0 & 1 \\ 0 & 1 & \ldots & 0 & 1 \\ . & . & . & \ldots & . \\ 0 & 0 & \ldots & 1 & 1 \end{pmatrix}$$

Удобно представлять матрицу G в виде $(E_k \mid A)$.

Тогда исходное сообщение $u = (u_1, u_2, \ldots, u_k)$ переводится (добавляются проверочные символы) в (u, uA).

Для однозначного кодирования важно, чтобы из условия $u \neq u'$ всегда следовало, что

$$uG \neq u'G.$$

Это эквивалентно (*линейная алгебра!*) тому, что строки матрицы G линейно независимы. Таким образом, строки матрицы G образуют базис линейного пространства C.

Линейный код можно задать и как решение систем однородных линейных уравнений

$$C = \{x \in K^n \mid xH^t = 0,\ H - (n-k) \times n \text{ матрица}\}.$$

Определение 2.3. Матрица H называется *проверочной матрицей* кода C, если выполнено условие: $x \in C$ тогда и только тогда, когда

$$xH^t = 0.$$

Проверочная и порождающая матрицы связаны соотношением

$$GH^t = 0.$$

Если $G = (-A^t|E_k)$, то

$$H = (E_{n-k}|A).$$

2.2 Кодовое расстояние линейного кода

Теорема 2.2. Если любые $s \leq d-1$ столбцов проверочной матрицы H линейного (n,k)-кода линейно независимы, то минимальное расстояние кода равно по меньшей мере d. Если при этом найдутся d линейно зависимых столбцов, то минимальное расстояние кода равно d в точности.

Теорема 2.3. Если минимальное расстояние линейного (n,k)-кода равно d, то любые $l \leq d-1$ столбцов проверочной матрицы H линейно независимы и найдутся d линейно зависимых столбцов.

2.3 Декодирование линейного кода

Для линейных кодов проверка на отсутствие ошибок сводится к умножению на проверочную матрицу.

Исправление ошибок осуществляется следующим образом. Для принятого вектора x вычисляется

$$xH^t.$$

Пусть $e = (e_1, e_2, \ldots, e_n)$ – вектор ошибок, $c = (c_1, c_2, \ldots, c_n)$ – сообщение, которое было послано. Тогда $x = e + c$ и

$$xH^t = (e+c)H^t = eH^t.$$

Введем отношение эквивалентности на множестве K^n. Векторы x и y эквивалентны, если

$$x - y \in C.$$

Два вектора x и y эквивалентны, если

$$(x-y)H^t = 0.$$

Рассмотрим класс эквивалентных элементов $x + C = \{x + c \mid c \in C\}$. Синдромом этого класса называется элемент $s \in x + C$ наименьшего веса (их может быть несколько, тогда выбираем произвольный из них).

Если запомнить все синдромы и из произведение на матрицу H^t, то исправление ошибки сводится к следующей процедуре:

1. Вычислить $r = xH^t$.

2. Если $r = 0$, то ошибок не было. Иначе найти из таблицы синдром s, соответствующий r и вернуть $x - s$.

2.4 Примеры. Код Хемминга, код с проверкой на четность

Рассматриваем бинарный код. Код с проверочной матрицей, у которой в качестве столбцов берутся все двоичные представления чисел от 1 до m называется кодом Хемминга.

Пример.

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

проверочная матрица для $[2^3 - 1, 2^3 - 1 - 3, 3] = [7, 4, 3]$ кода.

В случае одной ошибки синдром eH^t совпадает с номером позиции, в которой произошла ошибка.

2.5 Построение линейных кодов в Sage

Построить линейные коды в Sage можно различными способами.

1. Задание с помощью порождающей матрицы.

In [3]:
```
#Определим порождающую матрицу над полем GF(2)
M = matrix(GF(2), [[1, 0, 1, 1, 0],\
[0, 1, 0, 1, 1]])
#Создание линейного кода с порождающей матрицей G
C = LinearCode(M)
#Вывод всех  элементов кода
C.list()
```

Out [3]: $[(0, 0, 0, 0, 0), (1, 0, 1, 1, 0), (0, 1, 0, 1, 1), (1, 1, 1, 0, 1)]$

В примере сначала вводится матрица над полем GF(2), а потом определяется линейный код с порождающей матрицей M.

Функции для работы с кодами находятся в пространстве имен codes. Создан объект C – линейный код, у которого через методы можно найти его параметры.

Можем вычислить порождающую и проверочную матрицу, кодовое расстояние, размерность длину кода.

```
In [4]: print('Порождающая матрица: ')
        print(C.generator_matrix())
        print('Проверочная матрица: ')
        print(C.parity_check_matrix())
        print('Кодовое расстояние = ', C.minimum_distance())
        print('Размерность = ', C.dimension())
        print('Длина = ', C.length())
```

```
Out [4]: Порождающая матрица:
         [1 0 0 1 0]
         [0 0 1 1 1]
         [0 1 0 1 1]
         Проверочная матрица:
         [1 0 0 1 1]
         [0 1 1 0 1]
         Кодовое расстояние = 2
         Размерность = 3
         Длина = 5
```

Иногда нам может понадобиться перебрать все элементы кода C. Мы можем для этого пройтись по всем элемента кода с помощью цикла.

В следующем фрагменте мы выводим все веса Хемминга для кода C.

```
In [5]: for c in C:
                    print(c, "wt=", c.hamming_weight())
```

```
Out [5]: (0, 0, 0, 0, 0) wt= 0
         (1, 0, 0, 1, 0) wt= 2
         (0, 0, 1, 1, 1) wt= 3
         (1, 0, 1, 0, 1) wt= 3
         (0, 1, 0, 1, 1) wt= 3
         (1, 1, 0, 0, 1) wt= 3
         (0, 1, 1, 0, 0) wt= 2
```

(1, 1, 1, 1, 0) wt= 4

2. Специальные коды

Наиболее популярные коды уже реализованы в Sage. В следующем примере задается $[7, 4]$-код Хемминга.

In [6]:
```
# Код Хемминга порядка m=3. Длина кода 2^m-1, размерность 2^m-1-m
C = codes.HammingCode(GF(2),3);C
```

Out [6]: [7, 4] Hamming Code over GF(2)

Напишем скрипт, выводящий параметры кода.

In [7]:
```
# Функция для вывода параметров кода
def printParameters(C):
    print('Порождающая матрица: ')
    print(C.generator_matrix())
    print('Проверочная матрица: ')
    print(C.parity_check_matrix())
    print('Кодовое расстояние = ', C.minimum_distance())
    print('Размерность = ', C.dimension())
    print('Длина = ', C.length())
```

Выведем теперь параметры, полученного кода.

In [8]:
```
printParameters(H)
```

Out [8]: Порождающая матрица:

[1 0 0 0 0 1 1]

[0 1 0 0 1 0 1]

[0 0 1 0 1 1 0]

[0 0 0 1 1 1 1]

Проверочная матрица:

[1 0 1 0 1 0 1]

[0 1 1 0 0 1 1]

[0 0 0 1 1 1 1]

Кодовое расстояние = 3

Размерность = 4

Длина = 7

Еще один пример кода

In [9]:
```
# Двоичный (расширенный) код Голея
C=codes.GolayCode(GF(2)); C
```

Out [9]: [24, 12, 8] Extended Golay code over GF(2)

Полный список доступных кодов можно найти в документации.

3. Кодирование и декодирование линейных кодов

Каждому линейному коду можно сопоставить кодировщик (encoder) и декодировщик (decoder). Рассмотрим как работают эти объекты.

In [10]:
```
# Построим новый код по порождающей матрице
G = matrix(GF(2), [[1, 0, 1, 1, 0], [0, 1, 0, 1, 1]]);
C=codes.LinearCode(G);
C.list()
```

Out [10]: [(0, 0, 0, 0, 0), (1, 0, 1, 1, 0), (0, 1, 0, 1, 1), (1, 1, 1, 0, 1)]

Найдем минимальное расстояние кода.

In [11]:
```
C.minimum_distance()
```

Out [11]: 3

Зададим кодировщик линейного кода.

In [12]:
```
#encoder кода C
Encoder=C.encoder(); Encoder
```

Out [12]: Generator matrix-based encoder for [5, 2] linear code over GF(2)

Воспользуемся кодировщиком для получения кодового слова из информационного сообщения.

In [13]:
```
# Зададим информационное сообщение
message=vector(GF(2), [1,1]);
# Закодируем информационное сообщение = message*G
word=Encoder(message); word
```

Out [13]: $(1, 1, 1, 0, 1)$

Сымитируем возникновение одной ошибки в кодовом слове при передаче по каналу связи.

In [14]:
```
#Зададим вектор ошибок
err_vect = vector(GF(2), (0, 0, 0, 0, 1))
#Слово, в котором произошла она ошибка
word_err = word+err_vect;
word_err
```

Out [14]: $(1, 1, 1, 0, 0)$

Создадим декодер.

In [15]:
```
# decoder кода C
Decoder=C.decoder(); Decoder
```

Out [15]: Syndrome decoder for [5, 2] linear code over GF(2) handling errors of weight up to 2

Применим декодер к слову без ошибок.

In [16]:
```
# Декодирование вектора без ошибок
Decoder.decode_to_code(word)
```

Out [16]: $(1, 1, 1, 0, 1)$

Применим декодер к слову с одной ошибкой.

```
In [17]: # Декдирование вектора с одной ошибкой
         Decoder.decode_to_code(word_err)
```

Out [17]: (1, 1, 1, 0, 1)

Декодер восстановил правильное слово.

Теперь получим исходное информационное сообщение.

```
In [18]: # Получение вектора информационных символов
         Decoder.decode_to_message(word_err)
```

Out [18]: (1,1)

Можем получить полный список синдромов и лидеров смежных классов. Для этого получим сначала объект - декодер на основе таблицы синдромов и лидеров смежных классов.

```
In [19]: # список синдромов и лидеров смежных классов
         D=codes.decoders.LinearCodeSyndromeDecoder(C);D
```

Out [19]: Syndrome decoder for [5, 2] linear code over GF(2) handling errors of weight up to 2

Выведем таблицу в виде синдром - лидер смежного класса.

```
In [20]: T=D.syndrome_table(); T
```

Out [20]: (0, 0, 0): (0, 0, 0, 0, 0),
(1, 0, 0): (1, 0, 0, 0, 0),
(0, 1, 0): (0, 1, 0, 0, 0),
(0, 0, 1): (0, 0, 1, 0, 0),
(1, 0, 1): (0, 0, 0, 1, 0),
(1, 1, 1): (0, 0, 0, 0, 1),
(1, 1, 0): (1, 1, 0, 0, 0),
(0, 1, 1): (1, 0, 0, 0, 1)

Для демонстрации выведем все смежные классы по линейному пространству C.

Сначала опишем вспомогательную функцию.

In [21]:
```
# Функция для вывода членов смежного класса вектора v по C
def cosetList(C, v):
    l=[]
    v=vector(GF(2), v)
    for x in C.list():
        l.append(v+x)
    return l
```

Применим эту функцию для вывода всех смежных классов

In [22]:
```
for s in T:
    print("Синдром: ", s)
    print("Лидер класса: ", T[s])
    print("Члены смежного класса: ", cosetList(C, T[s]), "\n")
```

Out [22]:
Синдром: (0, 0, 0)
Лидер класса: (0, 0, 0, 0, 0)
Члены смежного класса: [(0, 0, 0, 0, 0), (1, 0, 1, 1, 0), (0, 1, 0, 1, 1), (1, 1, 1, 0, 1)]

Синдром: (1, 0, 0)
Лидер класса: (1, 0, 0, 0, 0)
Члены смежного класса: [(1, 0, 0, 0, 0), (0, 0, 1, 1, 0), (1, 1, 0, 1, 1), (0, 1, 1, 0, 1)]
...
Синдром: (1, 1, 1)
Лидер класса: (0, 0, 0, 0, 1)
Члены смежного класса: [(0, 0, 0, 0, 1), (1, 0, 1, 1, 1), (0, 1, 0, 1, 0), (1, 1, 1, 0, 0)]
...
Синдром: (0, 1, 1)
Лидер класса: (1, 0, 0, 0, 1)
Члены смежного класса: [(1, 0, 0, 0, 1), (0, 0, 1, 1, 1), (1, 1, 0, 1, 0), (0, 1, 1, 0, 0)]

В этом примере можно заметить, что в последней строке вес лидера смежного класса равен двум и внутри смежного класса два элемента с таким весом. Это согласуется с тем, что кодовое расстояние равно 3 и код может исправлять только одну ошибку.

2.6 Упражнения

Все коды рассматриваются над полем $F = GF(2)$. Для вычислений используйте Sage.

2.1. Пусть $C = \{000000, 100110, 010101, 001011, 101101, 011110, 110011, 111000\}$. Зная, что это линейный код, найдите минимальное кодовое расстояние. Можете ли вы доказать, что это действительно линейный код? Найдите базисные векторы.

2.2. Найдите проверочную и порождающую матрицу линейного кода

$$C = \{(x_1, x_2, x_3, x_4) \mid x_1 + x_2 + x_3 + x_4 = 0\}.$$

Найдите кодовое расстояние кода из проверочной матрицы, число ошибок, которые код обнаруживает и исправляет. Найдите все синдромы кода и декодируйте $(1, 1, 1,)0$.

2.3. Пусть

$$C = \{000000, 101110, 001010, 110111, 100100, 011001, 111101, 010011\}$$

Найдите порождающую и проверочную матрицу линейного кода.

2.4. Выпишите все кодовые слова двоичного кода, заданного порождающей матрицей. Найдите проверочную матрицу линейного кода.

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

Найдите кодовое расстояние кода из проверочной матрицы, число ошибок, которые код обнаруживает и исправляет. Найдите все синдромы кода. Декодируйте 1111 и 0101.

2.5. Найдите проверочную матрицу линейного кода с порождающей матрицей

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

Найдите кодовое расстояние кода из проверочной матрицы, число ошибок, которые код обнаруживает и исправляет. Найдите все синдромы кода. Декодируйте 11111 и 01011.

2.6. Для данного множества $V \subset B^n$ найти (минимальное) кодовое расстояние. Для каждого из кодов найти: число ошибок, которые код обнаруживает и исправляет.

1) $V = \{11000, 10101, 01110\}$,

2) $V = \{111100, 110011, 001111\}$.

2.7. Найдите все слова, которые находятся на расстоянии 3 от слова 11000 в $\mathbb{F}_2^5$.

2.8. Рассмотрим $(6, 8, 3)_2$ линейный код

$$C = \{000000, 100110, 010101, 001011, 110011, 101101, 011110, 111000\}$$

с порождающими $100110, 010101, 001011$.

Покажите, что кодовое расстояние равно трем. Восстановите сообщения $111100, 111011$.

2.9. Выпишите все элементы линейного кода с порождающей матрицей

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

Найдите кодовое расстояние кода, число ошибок, которые код обнаруживает и исправляет.

2.10. Выпишите все элементы линейного кода с порождающей матрицей

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

Найдите кодовое расстояние кода, число ошибок, которые код обнаруживает и исправляет.

2.11. Используя бинарный код Хэмминга $[7, 4, 3]_2$, декодируйте сообщения $y_1 = (1, 1, 0, 1, 1, 0, 0)$.

2.12. Используя бинарный код Хэмминга $[7, 4, 3]_2$, декодируйте сообщения $y_2 = (1, 1, 1, 1, 1, 1, 1)$, $y_3 = (1, 1, 1, 0, 0, 0, 0)$.

2.13. Пусть C_1 и C_2 линейные коды одинаковой длины n с порождающими матрицами G_1 и G_2 и размерностями $k_1 \geq k_2 > 0$. Определим следующие (не обязательно линейные) коды: $C_3 = C_1 \cup C_2$; $C_4 = C_1 \cap C_2$; $C_5 = C_1 + C_2 = \{x + y \mid x \in C_1, y \in C_2\}$; $C_6 = \{(x, y) \mid x \in C_1, y \in C_2\} \subset F^{2n}$.

1. Покажите, что коды C_4, C_5, C_6 – линейные.
2. При каких условиях код C_3 является линейным?
3. Докажите, что кодовое расстояние $d(C_4) \geq \max(d(C_1), d(C_2))$.
4. Выразите порождающую матрицу кода C_6 через матрицы G_1 и G_2.
5. Докажите, что кодовое расстояние $d(C_6) = \min(d(C_1), d(C_2))$.

Глава 3

ГРАНИЦЫ ОБЪЕМОВ КОДОВ

В этой главе мы приведем некоторые простые оценки, связывающие параметры кода.

Пусть F – алфавит из q элементов, C – (n, M, d)-код. Напомним, что это означает, что длина кодовых слов равна n, число элементов в C равно M (мы также будем писать $|C|$), минимальное расстояние между словами равно d. В этом случае код C может обнаруживать $d-1$ ошибку и исправлять $t = [\frac{d-1}{2}]$ ошибок.

3.1 Граница сферической упаковки

Обозначим через

$$B_r(x) = \{y \in F^n \mid d(x, y) \leq r\}$$

шар радиуса r.

Лемма 3.1. Количество элементов в $B_r(x)$ равно

$$|B_r(x)| = \sum_{k=0}^{r} C_n^k (q-1)^k.$$

Действительно. Пусть x – слово длины n. Подсчитаем сколько векторов y имеет расстояние $d(x, y) = k$. Существует ровно C_n^k способов выбрать позиции $i_1, i_2, \ldots, i_k$, в которых слова x и y отличаются. Если позиции зафиксированы, то существует $(q-1)^k$ способов поставить элементы отличные от $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$.

Отметим также, что количество элементов в шаре $B_t(x)$ не зависит от центра. Обозначим $|B_t| = |B_t(x)|$ для произвольного x.

Теорема 3.1. (Граница Хэмминга, граница сферической упаковки) Пусть C – (n, M)-код, исправляющий t ошибок. Тогда

$$|C| \leq \frac{q^n}{\sum\limits_{k=0}^{t} C_n^k (q-1)^k}.$$

Доказательство. Так как код C исправляет t ошибок, то шары радиуса t с центрами в точках из C не пересекаются. Таким образом $\bigcup\limits_{x\in C} B_t(x)$ состоит из различных элементов и количество этих элементов не превосходит q^n. Итак

$$\bigcup_{x\in C} B_t(x) = |C||B_t| \leq q^n.$$

Применяя лемму, получим

$$|C| \leq \frac{q^n}{|B_t|} = \frac{q^n}{\sum\limits_{k=0}^{t} C_n^k (q-1)^k}.$$

□

Определение 3.1. Код, для которого достигается равенство, называется *совершенным*.

Пример 3.1. Код Хемминга является совершенным.

Действительно, код Хемминга является $[2^m-1, 2^m-1-m, 3]_2$-кодом и исправляет одну ошибку. Количество элементов в коде равно 2^{2^m-1-m}. Найдем $B_1(x) = C_n^0 + C_n^1(2-1) = 1+n = 1+2^m-1 = 2^m$. Имеем $2^{2^m-1-m} = 2^{2^m-1}/2^m$.

Замечание. Можно показать, что совершенные двоичные линейные коды исчерпываются кодами Голея, кодами Хемминга и кодом с повторением.

3.2 Граница Синглтона

Теорема 3.2. (Граница Синглтона) Для произвольного кода $C \subseteq F^n$ над алфавитом из q элементов с минимальным расстоянием d выполнено

$$|C| \leq q^{n-d+1}.$$

В частности, если код C линейный и $k = \dim C$, то

$$d \leq n - k + 1.$$

Доказательство. Для любого подмножества $C \subseteq F^n$ имеем $|C| \leq q^n$. Рассмотрим

$$C' = \{(c_d, c_{d+1}, \ldots, c_n) \mid \exists (c_1, c_2, \ldots, c_{d-1}, c_d, c_{d+1}, \ldots, c_n) \in C\}.$$

Так как минимальное расстояние между словами в C равно d, то все элементы в C' различны. Поэтому $|C| = |C'| \leq q^{n-(d-1)} = q^{n-d+1}$. □

Определение 3.2. Код, для которого выполняется равенство в оценке Синглтона, называется *кодом с максимально достижимым расстоянием* (МДР-кодом).

Пример 3.2. Код Хемминга не является МДР-кодом.

Пример 3.3. Код с повторениями и код проверки на четность являются МДР-кодами.

Пример 3.4. (обобщенный код Рида-Соломона) Пусть $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ различные элементы поля $F = GF(q)$. Обобщенным кодом Рида-Соломона назовем $[n, k, d]$ код с проверочной матрицей

$$H = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ \alpha_1 & \alpha_2 & \ldots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \ldots & \alpha_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{n-k-1} & \alpha_2^{n-k-1} & \ldots & \alpha_n^{n-k-1} \end{pmatrix}.$$

Естественно, $n < q$ и $k \leq n - 2$.

В этой матрице любые $n - k$ столбцов линейно независимы. Поэтому $d \geq n - k + 1$. С другой стороны из границы Синглтона $d \leq n - k + 1$. Таким образом, $d = n - k + 1$.

3.3 Граница Варшамова-Гилберта

Обозначим через

$$B_q(n, t) = \sum_{k=0}^{t} C_n^k (q-1)^k$$

количество элементов в шаре радиуса t в n-мерном пространстве.

Теорема 3.3 (Граница Варшамова-Гилберта)**.** Пусть $F = GF(q)$, n, k, d – положительные целые числа, для которых выполнено условие

$$B_q(n-1, d-2) < q^{n-k}.$$

Тогда существует линейный $[n, k, d]_q$-код.

Доказательство. Построим $n - k \times n$ матрицу H, у которой любые $d - 1$ столбец линейно независимы. Это даст нам линейный код с требуемым расстоянием d.

Первые $n-k$ столбцов $h_1, h_2, \ldots, h_{n-k}$ матрицы H составляют единичную матрицу. Остальные столбцы будем строить итеративно.

Пусть уже построены столбцы $h_1, h_2, \ldots, h_{s-1}$, удовлетворяющие свойству, что любые $d-1$ столбец линейно независимые. Необходимо найти вектор $h_s \in F^{n-k}$, такой, что в построенном наборе снова любые $d-1$ столбец линейно независимые.

Рассмотрим множество всех векторов вида

$$\{\alpha_1 h_1 + \alpha_2 h_2 + \ldots + \alpha_{s-1} h_{s-1}\},$$

где $\alpha_i \in F$, по крайней мере один из $\alpha_i = 0$.

Это те векторы, которые 'не подходят' для построения следующего столбца $h_s \in F^{n-k}$. Их количество не превосходит числа векторов вида $a = (\alpha_1, \alpha_2, \ldots, \alpha_{s-1})$, имеющих вес Хемминга $wt(a) \leq d-2$.

Так как

$$|\{a \mid wt(a) \leq d-2\}| = B_q(s-1, d-2) \leq B_q(n-1, d-2) < q^{n-k},$$

то существует вектор h_s, удовлетворяющий условию. □

Упражнения.

3.1. Рассмотрите линейный код с порождающей матрицей

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

Является ли эти код совершенным или МДР-кодом?

3.2. Рассмотрите линейный код с порождающей матрицей

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

Является ли эти код совершенным или МДР-кодом?

3.3. Напишите функцию isPerfect() на языке Python, которая принимает на вход линейный код и проверяет является ли он совершенным.

Глава 4

БХЧ-КОДЫ И КОДЫ РИДА-СОЛОМОНА

В этой главе мы рассмотрим два примера кодов: БХЧ-коды и коды Рида-Соломона. Эти примеры служат обоснованием введения понятия циклических кодов, которые рассматриваются в следующей главе.

4.1 Коды Боуза-Хоквингема-Чоудхури

Построим код, который умеет исправлять заданное число ошибок, и количество проверочных символов будет порядка $\log(n)$, где n – длина кодовых слов.

Пусть в качестве алфавита выбрано поле $F = GF(q)$. Зафиксируем m и построим код с минимальным расстоянием $d = 2t + 1$ длины $n = q^m - 1$.

Рассмотрим расширение E поля F порядка m. Пусть β – примитивный элемент поля E. Тогда все элементы

$$\beta, \beta^2, \beta^3, \ldots, \beta^{2t}$$

различны. Обозначим

$$h_1, h_2, \ldots, h_{2t}$$

минимальные многочлены этих элементов над полем $GF(q)$.

Пусть теперь

$$g(x) = \mathrm{lcm}(h_1, h_2, \ldots, h_{2t})$$

наименьшее общее кратное этих многочленов.

Будем кодировать сообщения по следующей схеме:

$$(u_0, u_1, \ldots, u_{k-1}) \to u_0 + u_1 x g(x) + \ldots + u_{k-1} x^{k-1} g(x) \mod (x^n - 1)$$

Здесь мы используем соответствие между векторами вида $(c_0, c_1, \ldots, c_{n-1})$ и многочленами $c(x) = c_0 + c_1 x + \ldots + c_{n-1} x^{n-1}$.

Таким образом, мы можем закодировать слова (многочлены) степени $k \leq n - \deg g$. Отметим, что степени h_i не превышают m, поэтому $k \geq n - 2tm$.

Теорема 4.1. Для построенного кода C кодовое расстояние больше или равно d.

Доказательство. Слово $c \in C$ тогда и только тогда, когда $c(x)$ аннулируется элементами $\beta, \beta^2, \beta^3, \ldots, \beta$

Запишем это в матричном виде

$$\begin{pmatrix} 1 & \beta & \beta^2 & \ldots & \beta^{(n-1)} \\ 1 & \beta^2 & \beta^{2(2)} & \ldots & \beta^{(n-1)2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \beta^{2t} & \beta^{2(2t)} & \ldots & \beta^{(n-1)(2t)} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix} = 0.$$

Предположим, что $w = wt(c) \leq d - 1$. Это означает, что существует индексы $a_1, a_2, \ldots, a_w$, такие, что $c_i \neq 0$, только для $i \in \{a_1, a_2, \ldots, a_w\}$.

Тогда выделяя подходящие столбцы матричное равенство можно записать в виде

$$\begin{pmatrix} \beta^{a_1} & \beta^{a_2} & \ldots & \beta^{a_w} \\ \beta^{a_1(2)} & \beta^{a_2(2)} & \ldots & \beta^{a_w(2)} \\ \vdots & \vdots & \vdots & \vdots \\ \beta^{a_1(2t)} & \beta^{a_2(2t)} & \ldots & \beta^{a_w(2t)} \end{pmatrix} \begin{pmatrix} c_{a_1} \\ c_{a_2} \\ \vdots \\ c_{a_w} \end{pmatrix} = 0.$$

Так как по предположению $w \leq d-1$, то получим СЛУ с квадратной матрицей и определитель этой матрицы не равен нулю (надо вынести из каждого столбца $\beta_{a_i} l$ и получим определитель Вандермонда). Значит система может иметь только нулевое решение – противоречие. □

Коды, построенные таким способом, называются кодами Боуза-Хоквингема-Чоудхури (БХЧ-кодами).

4.2 Код Рида-Соломона

Пусть $F = GF(q)$ – конечное поле, $\alpha_1, \alpha_2, \ldots, \alpha_n \in F$ и $k \leq n$.

Определение 4.1. Кодом Рида-Соломона называется линейный код $C \subseteq F^n$ с порождающей матрицей

$$G = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ \alpha_1 & \alpha_2 & \ldots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \ldots & \alpha_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^k & \alpha_2^k & \ldots & \alpha_n^k \end{pmatrix}.$$

Если $u = (u_0, u_1, \ldots, u_{k-1})$ – информационное сообщение и $u(x) = u_0 + u_1 x + \ldots + u_{k-1}x^{k-1}$ соответствующий ему многочлен, то кодовое слово вычисляется по формуле

$$uG = (u(\alpha_1), u(\alpha_2), \ldots, u(\alpha_n)).$$

Так как многочлен степени не выше $k-1$ имеет не больше $k-1$ корней, то различные кодовые слова могут имеют расстояние $d \geq n - (k-1)$. Учитывая оценку Синглтона, получим, что код Рида-Соломона является МДР кодом, то есть

$$d(C) = n - k + 1.$$

Пусть $d = 2t + 1$, то есть код может исправлять t ошибок, тогда $n = k + 2t$.

Декодирование кода Рида-Соломона.

Опишем основную идею декодирования кодов Рида-Соломона.

Пусть $u(x) = u_0 + u_1 x + \ldots + u_{k-1}x^{k-1}$ полином степени меньше k и передана таблица значений $u(\alpha_1), u(\alpha_2), \ldots, u(\alpha_n)$, в которых может произойти не более чем t ошибок. Требуется восстановить $u(x)$. Обозначим полученные значения $\widetilde{u}(\alpha_1), \widetilde{u}(\alpha_2), \ldots, \widetilde{u}(\alpha_n)$

Отметим, что если бы ошибок не было, то взяв любые k значений $u(\alpha_{i_1}), u(\alpha_{i_2}), \ldots, u(\alpha_{i_k})$ можно восстановить $u(x)$ построив интерполяционный многочлен по k точкам.

Предположим ошибки произошли в точках $\beta_1, \beta_2, \ldots, \beta_s$, $s \leq t$ (отметим, что в реальности нам эти позиции не известны). Существует многочлен $D(x)$ степени t, старший коэффициент которого равен 1 и который обращается в ноль в точках $\beta_1, \beta_2, \ldots, \beta_s$. Например, это может быть многочлен вида $(x - \beta_1)(x - \beta_2)\ldots(x - \beta_s)x^{t-s}$.

Рассмотрим многочлен

$$Q(x) = u(x)D(x).$$

Он будет обладать следующим свойством:

$$Q(\alpha_i) = \widetilde{u}(\alpha_i)D(\alpha_i) = u(\alpha_i)D(\alpha_i).$$

Лемма 4.1. Существуют многочлены $\widetilde{Q}(x)$ и $\widetilde{D}(x)$, такие что $\deg \widetilde{D}(x) \leq t$, $\deg \widetilde{Q}(x) < k + t$ и

$$\widetilde{Q}(\alpha_i) = \widetilde{u}(\alpha_i)\widetilde{D}(\alpha_i), \qquad i = 1, 2, \ldots, n.$$

Доказательство. Имеем систему однородных линейных уравнений с менее чем $t + k + t = n$ неизвестными (коэффициентами многочленов) и n уравнениями. Разрешая эту систему и выбирая ненулевое решение, получим требуемые многочлены. □

Лемма 4.2. Пусть $\widetilde{Q}(x)$ и $\widetilde{D}(x)$ – многочлены, условиям предыдущей леммы. Тогда $\widetilde{Q}(x)$ делится на $\widetilde{D}(x)$ и

$$\widetilde{Q}(x)/\widetilde{D}(x) = u(x).$$

Доказательство. Многочлены $u(x)\widetilde{D}(x)$ и $\widetilde{Q}(x)$ имеют степень меньше $t+k$ и совпадают в не менее чем $n-t$ точках. Так как $n-t = k+2t-t = k+t$, то $u(x)\widetilde{D}(x) = \widetilde{Q}(x)$. □

Упражнения

4.1. Рассмотрите расширение $GF(8)$ поля $GF(2)$. Постройте БХЧ код, исправляющий:

a) одну ошибку;

b) две ошибки;

c) три ошибки.

Глава 5

ЦИКЛИЧЕСКИЕ КОДЫ

Циклические коды обладают дополнительной алгебраической структурой. Это позволяет организовать кодирование и декодирование более простыми способами, чем общая конструкция линейного кода. Среди циклических кодов достаточно много кодов с хорошими свойствами.

Определение 5.1. Линейный код $C \subseteq F^n$ называется циклическим, если из того, что

$$c = (c_0, c_1, \ldots, c_{n-2}, c_{n-1})$$

принадлежит коду C, следует, что вектор (слово в алфавите F)

$$(c_{n-1}, c_0, c_1, \ldots, c_{n-2})$$

также принадлежит коду C. Это кодовое слово называется *циклическим сдвигом* слова c.

Пример 5.1. Код $C_1 = \{(0,0,0,0), (0,1,0,1), (1,0,1,0), (1,1,1,1)\}$ является циклическим, а код $C_2 = \{(0,0,0,0), (1,0,0,1), (0,1,1,0), (1,1,1,1)\}$ не является.

Пример 5.2. Код с повторением и двоичный код проверки на четность – циклические.

Циклические коды можно рассматривать как идеалы в кольце $F[x]/(x^n - 1)$ или идеалы в групповой алгебре FC_n над циклической группой C_n.

5.1 Циклические коды как идеалы $F[x]/(x^n - 1)$

Векторное пространство F^n можно представлять в разных формах. Например, это пространство естественным образом изоморфно пространству V_n всех многочленов из $F[x]$, степени которых строго меньше n. Изоморфизм осуществляется следующим образом:

$$c = (c_0, c_1, \ldots, c_{n-1}) \leftrightarrow c(x) = \sum_{i=0}^{n-1} c_i x^i.$$

В дальнейшем мы будем по мере необходимости переходить от одного из этих пространств к другому, используя описанный выше изоморфизм.

Еще одним представлением пространства F^n является факторкольцо кольца многочленов по идеалу, порожденному многочленом $x^n - 1$.

Отождествим множество полиномов степени строго не выше n с факторкольцом

$$S = F[x]/(x^n - 1)$$

кольца многочленов $F[x]$ по идеалу, порожденному $x^n - 1$.

Будем обозначать через $\overline{f(x)}$ образ элемента $f(x) \in F[x]$ в S при естественном эпиморфизме.

Наше отождествление F^n и V_n может быть продолжено до отождествления F^n и S.

$$v = (v_0, v_1, \ldots, v_{n-1}) \leftrightarrow \overline{v} = \sum_{i=0}^{n-1} v_i \overline{x}^i.$$

Легко видеть, что умножение полинома в S на $\overline{x}$ эквивалентно циклическому сдвигу соответствующего вектора в F^n.

Учитывая вышесказанное, справедливо еще одно определение циклического кода:

Определение 5.2. Код C (как подмножество в S) называется циклическим, если C является идеалом в кольце S.

Замечание. С точки зрения вычислений факторкольцо $F[x]/(x^n-1)$ выглядит как множество многочленов степени не выше чем $n-1$ с обычной операцией сложения многочленов, а умножение производится по модулю многочлена $x^n - 1$, то есть сначала надо перемножить многочлены обычным способом, а потом взять остаток от деления на $x^n - 1$.

Далее мы будем отождествлять $\overline{g(x)}$ и его прообраз в $F[x]$ для обозначения элементов из S и опускать значок верхнего подчеркивания.

Итак, циклические коды находятся во взаимно-однозначном соответствии с идеалами факторалгебры $F[x]/(x^n - 1)$ и могут быть описаны на языке многочленов.

Так как кольцо многочленов $F[x]$ является кольцом главных идеалов, то все идеалы S являются главными и порождаются многочленами $\overline{g(x)}$, где $g(x)$ делит $x^n - 1$.

Более точно строение порождающих многочленов описывается следующей теоремой.

Теорема 5.1. Пусть C – циклический код в $F[x]/(x^n - 1)$. Тогда существует (и единственный) многочлен $g(x)$, порождающий идеал C, такой, что его степень $r < n$ минимальна и коэффициент при старшей степени равен 1.

Многочлен $g(x)$ делит $x^n - 1$ и любой элемент из C может быть записан в виде произведения $g(x)a(x)$, где степень $a(x)$ меньше $n - r$.

Пример 5.3. Циклические бинарные коды длины 7.

Рассмотрим все простые делители многочлена x^7-1 в кольце многочленов $GF(2)[x]$. Используя Sage, получим разложение многочлена на простые множители.

```
In [23]: # Определим кольцо многочленов над конечным полем
         R = PolynomialRing(GF(2),'x')
         # Нужно определить переменную x как объект этого кольца
         x = R.gen()
         # Запишем многочлен, который будем раскладывать
         f = x^7-1
         # Операция разложения на неприводимы множители
         f.factor()
Out [23]: (x + 1) * (x^3 + x + 1) * (x^3 + x^2 + 1)
```

Итак,

$$x^7-1=(x+1)*(x^3+x+1)*(x^3+x^2+1).$$

Перебирая все делители многочлена x^7-1 мы можем построить $C_3^1+C_3^2=3+3=6$ кодов.

Выпишем, например, циклический код, порожденный $(x^3+x+1)*(x^3+x^2+1)=x^6+x^5+x^4+x^3+x^2+x+1$:

$$C=\{0,x^6+x^5+x^4+x^3+x^2+x+1\}$$

Этот код имеет размерность 1 и состоит из двух кодовых слов.

Теорема 5.2. Пусть C — циклический код, порожденный многочленом $g(x)=g_0+g_1x+\cdots+g_rx^r$. Тогда код C имеет порождающую матрицу размера $(n-r)\times n$

$$G=\begin{pmatrix} g_0 & g_1 & g_2 & \dots & g_r & 0 & 0 & \dots & 0 \\ 0 & g_0 & g_1 & \dots & \dots & g_r & 0 & \dots & 0 \\ 0 & 0 & g_0 & \dots & \dots & \dots & g_r & \dots & 0 \\ & & & \ddots & & & & \ddots & \\ 0 & 0 & 0 & \dots & g_0 & \dots & \dots & \dots & g_r \end{pmatrix}$$

и его размерность равна

$$\dim(C)=n-r$$

Доказательство. Заметим, что g_0 отличен от нуля. Так как иначе вектор $(0, g_1, \ldots, g_r, 0, \ldots, 0)$ содержится в C, следовательно и вектор $(g_1, \ldots, g_r, 0, \ldots, 0)$, полученный сдвигом, принадлежит C.

Учитывая, соответствие векторов многочленам в S, получим, что мы нашли ненулевой многочлен в C меньшей степени чем $g(x)$. Это противоречит минимальности степени образующего многочлена.

Таким образом, ранг матрицы G равен $n - r$. Осталось показать, что G – порождающая матрица.

Если $c(x) = a(x)g(x) = (a_0 + a_1x + a_2x^2 + \ldots + a_{n-r-1}x^{n-r-1})(g_0 + g_1x + \cdots + g_rx^r)$ кодовый многочлен в C, то

$$c(x) = a_0g(x) + a_1xg(x) + a_2x^2g(x) + \ldots + a_{n-r-1}x^{n-r-1}g(x).$$

Многочлену $x^kg(x)$ соответствует $k + 1$ строка матрицы G. Таким образом, все элементы C являются линейной комбинацией строк матрицы G.

Определение 5.3. Пусть C — циклический код с порождающим многочленом $g(x)$ и

$$x^n - 1 = g(x)h(x),$$

тогда $h(x)$ называется *проверочным многочленом* кода C.

Теорема 5.3. Пусть C есть циклический код с проверочным многочленом

$$h(x) = h_0 + h_1x + \ldots + h_kx^k,$$

Тогда проверочная матрица кода C имеет вид

$$H = \begin{pmatrix} h_k & h_{k-1} & h_{k-2} & \ldots & h_0 & 0 & 0 & \ldots & 0 \\ 0 & h_k & h_{k-1} & h_{k-2} & \ldots & h_0 & 0 & \ldots & 0 \\ 0 & 0 & h_k & h_{k-1} & h_{k-2} & \ldots & h_0 & \ldots & 0 \\ & & & \ddots & & & & \ddots & \\ 0 & 0 & 0 & \ldots & h_k & h_{k-1} & h_{k-2} & \ldots & h_0 \end{pmatrix}$$

Упражнение 5.1. Докажите утверждение.

Пример 5.4. Пусть C – бинарный циклический код длины 7 с порождающим многочленом $g(x) = x^3 + x + 1$. Тогда $h(x) = (x^7 - 1)/(x^3 + x + 1) = x^4 + x^2 + x + 1$ проверочный многочлен.

Порождающая матрица

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

Проверочная матрица

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

Заметим, что это $[7, 4, 3]$ код Хемминга.

Справедлива следующая теорема.

Теорема 5.4. Двоичный код Хемминга эквивалентен циклическому коду

Доказательство. Рассмотрим неприводимый многочлен $p(x)$ степени m над полем $\mathbb{F}_2 = GF(2)$.

Факторкольцо $\mathbb{F}_2[x]/p(x)$ полиномов по модулю $p(x)$ является полем и содержит 2^m элементов. Из теории полей известно, что существует элемент α поля $GF(2^m) = \mathbb{F}_2/p(x)$ такой, что

$$\mathbb{F}_2[x]/p(x) = \{0, 1, \alpha, \alpha^2, \dots \alpha^{2^m-2}\}.$$

Этот элемент называется примитивным элементов и его порядок равен $2^m - 1$.

Рассмотрим матрицу

$$\widetilde{H} = (1, \alpha, \alpha^2, \dots \alpha^{2^m-2})$$

над полем $\mathbb{F}_2[x]/p(x)$.

Так как поле $\mathbb{F}_2[x]/p(x)$ является m-мерным векторным пространством над полем $\mathbb{F}_2$ с базисом $1, x, \dots, x^{m-1}$, то элементы α^i представимы в виде

$$\alpha^i = a_{i,0} + a_{i,1}x + \dots + a_{i,m-1}x^{m-1},$$

где $a_{ik} \in \mathbb{F}_2$.

Обозначим $n = 2^m - 1$ и рассмотрим матрицу

$$H = \begin{pmatrix} a_{0,0} & a_{1,0} & a_{2,0} & \dots & a_{n-1,0} \\ a_{0,1} & a_{1,1} & a_{2,1} & \dots & a_{n-1,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{0,m-1} & a_{1,m-1} & a_{2,m-1} & \dots & a_{n-1,m-1} \end{pmatrix}.$$

H это $m \times (2^m - 1)$ матрица над полем $\mathbb{F}_2$ и она является в некотором смысле представлением матрицы $\widetilde{H}$.

Рассмотрим теперь линейный код C с проверочной матрицей H.

Сначала убедимся, что это код Хемминга. Действительно, так как все элементы α^i различны, то все столбцы матрицы H различные и ненулевые. Так как их $2^m - 1$, то это в точности все ненулевые столбцы высоты m.

Покажем, что C циклический код. Элемент $c = (c_0, c_1, \ldots, c_{n-1}) \in C \leftrightarrow Hc^t = 0$.

Матричное равенство $Hc^t = 0$ эквивалентно тому, что полином $c(x)$ обращается в ноль при подстановке $x = \alpha$ внутри $\mathbb{F}_2[x]/p(x)$. Итак получили критерий принадлежности слова коду C:

$$C = \{(c_0, c_1, \ldots, c_{n-1}) \mid c_0 + c_1\alpha + \ldots c_{n-1}\alpha^{n-1} = 0\} = \{c(x) \in \mathbb{F}_2[x]/(x^n - 1) \mid c(\alpha) = 0 \in \mathbb{F}_2[x]/p(x)\}.$$

Теперь мы можем доказать, то C – идеал в $\mathbb{F}_2[x]/(x^n - 1)$, а значит C – циклический код. Пусть $c(x) \in$, $f(x) \in \mathbb{F}_2[x]/(x^n - 1)$. Очевидно, что

$$c(\alpha)f(\alpha) = 0$$

и, поэтому, $c(x)f(x) \in C$. □

5.2 Sage

Как мы установили в лекции все циклические коды длины n над полем F порождаются многочленами – делителями многочлена $x^n - 1$.

Разложение многочлена $x^n - 1$ над конечным полем F на неприводимые сомножители в Sage можно произвести с помощью функции factor.

Разложим, например, многочлен $f = x^4 + x^2 - 1$ над полем $GF(9)$.

In [1]:
```
#Определим GF(9)
S.<z>=GF(9)
# Определим кольцо многочленов над конечным полем
R = PolynomialRing(S,'x')
# Нужно определить переменную x как объект этого кольца
x = R.gen()
# Запишем многочлен, который будем раскладывать
f = x^4+x^2-1
# Операция разложения на неприводимы множители
f.factor()
```

Out [1]: $(x\hat{}2 + z) * (x\hat{}2 + 2 * z + 1)$

Здесь элемент z это элемент поля $GF(9)$, построенного как факторкольцо кольца многочленов $\mathbb{Z}_3[z]$ по неприводимому многочлену второй степени.

Упражнения

5.1. Рассмотрим линейные коды длины $n = 7$.

(a) Сколько циклических кодов можно построить с такой длиной слова? Найти все порождающие полиномы.

(b) Может ли $g(x) = 1 + x^3 + x^4$ быть порождающим полиномом циклического (7, 3) кода?

(c) Сформируйте систематическую порождающую матрицу для кода с порождающим полиномом $g() = 1 + x$.

5.2. Описать строение всех циклических $[7, 4]_2$ кодов. Указание: найти разложение многочлена $x^7 - 1$ над полем $\mathbb{Z}_2$. Найти среди них многочлены степени $3 = 7 - 4$.

5.3. Показать, что код с проверкой на четность является циклическим. Указание: рассмотреть многочлены $g(x) = x - 1$ и $h(x)x^{n-1} + \ldots + x + 1$.

5.4. Закодировать в циклическом коде комбинации 1001, 1010, если образующий многочлен $g(x) = x^3 + x + 1$.

5.5. По заданному образующему многочлену $g(x) = x^4 + x^3 + 1$ построить образующую и проверочную матрицы.

Глава 6

ГРУППОВЫЕ КОДЫ

6.1 Определение групповых кодов

Эта глава посвящена групповым кодам. Подразумевается, что материал предназначен для работы на семинарах и будет разбираться студентами самостоятельно. Студентам будет необходимо разобрать статьи, выполнить вычисления в Sage. В конце главы приведены темы исследований, которые могут лечь в основу выпускной работы.

Пусть $F = GF(q)$ – конечное поле, G – конечная группа. Групповой алгеброй FG называется множество формальных сумм

$$FG = \left\{ \sum_{g \in G} \alpha_g g \mid \alpha \in F, g \in G \right\}.$$

Сложение и умножение в FG определяется естественным образом:

$$\sum \alpha_g g + \sum \beta_g g = \sum (\alpha_g + \beta_g) g,$$

$$(\sum \alpha_g g)(\sum \beta_g g) = \sum_g (\sum_h \alpha_{gh^{-1}} \beta_h) g.$$

Элементы $\sum \alpha_g g$ и $\sum \beta_g g$ равны тогда и только тогда, когда $\alpha_g = \beta_g$ для всех $g \in G$.

Определение 6.1. Будем называть групповым кодом любой идеал групповой алгебры FG.

Определение 6.2. Если группа G является абелевой, то групповые коды называются абелевыми кодами.

Пример. Если группа $G =< a \mid a^n = 1 >$ является циклической порядка n, то в силу изоморфизма

$$FG \cong F[x]/ < x^n - 1 >,$$

групповые коды совпадают с циклическими.

Задача. Представляет интерес рассмотрение односторонних идеалов вместо двусторонних в качестве кодов.

Исследования групповых кодов начались с работ Бермана, опубликованных в 1967 году ([6], [7]), который рассматривал абелевы коды.

В последнее время активизировались исследования для некоммутативных групп, в частности, получены описания для групповых алгебр над группами диэдра (см., например, [2], [3]).

В этой главе мы рассмотрим алгебраическое описание групповых кодов в случае когда FG является классически полупростой алгеброй.

Напомним одну из основополагающих теорем для полупростых групповых алгебр.

Теорема 6.1. (Машке) Пусть G – конечная группа, $|G| = n$, F – поле и его характеристика равна p. Групповая алгебра FG является классически полупростой тогда и только тогда, когда либо $p = 0$ либо $(n, p) = 1$.

В этом случае $FG \cong \bigoplus_{i=1}^{k} M_{n_i}(T_i)$, где T_i – алгебры с делением над F.

Так как групповая алгебра FG конечна, то мы можем применить теорему Веддерберна.

Теорема 6.2. (Веддерберн) Всякое конечное ассоциативное тело является полем.

Таким образом, если характеристика поля F взаимно проста с порядком группы, то

$$FG \cong \bigoplus_{i=1}^{k} M_{n_i}(F_i),$$

где F_i – конечные расширения поля F.

Как хорошо известно, любой идеал в классически полупростой алгебре является суммой неприводимых идеалов. Эти идеалы порождаются примитивными центральными идемпотентами и допускают непосредственные вычисления для групп и полей малых порядков.

Пусть G – конечная группа, $|G| = n$, F – поле.

6.1. Докажите, что $\mathbb{C}G \cong \oplus_{i=1}^{k} Mat_{n_i}(\mathbb{C})$ и $n = n_1^2 + n_2^2 + \ldots + n_k^2$.

6.2. Докажите, что $\mathbb{R}G \cong \oplus_{i=1}^{k} Mat_{n_i}(T_i)$, T_i – это либо тело кватернионов, либо $\mathbb{C}$, либо $\mathbb{R}$.

6.3. Докажите, что если $F = GF(q)$, то $FG \cong \bigoplus_{i=1}^{k} M_{n_i}(F_i)$, где F_i – конечные расширения поля F.

6.4. Докажите, что если $F = GF(q)$ и G – абелева, то существуют такие центральные идемпотенты $e_1, e_2, \ldots, e_n$, такие что $FG \cong \bigoplus_{i=1}^{k} (FG)e_i \cong F_i e_i$, где $F_i \cong (FG)e_i$ – конечные расширения поля F.

6.5. Пусть $F = GF(q)$, $(n, q) = 1$. Докажите, что многочлен $x^n - 1$ не имеет кратных корней (в поле разложения).

6.6. Докажите следующую теорему.

Теорема 6.3. Пусть $G =< g >$ циклическая группа порядка n над полем $GF(q)$ и $(n, q) = 1$. Пусть $x^n - 1 = (x-1)f_1(x)f_2(x)\dots f_k(x)$ разложение на неприводимые сомножители $x^n - 1$ над полем $GF(p)$. Тогда имеет место изоморфизм:

$$GF(p)G \cong GF(q) \oplus GF(q^{deg(f_1)}) \oplus GF(q^{deg(f_2)}) \dots \oplus GF(q^{deg(f_k)}).$$

Примените эту теорему для описания строения следующих групповых алгебр.

Упражнение 6.1. Найти описание групповой алгебры FG над полем $F = GF(2)$ циклической группы G порядка $k = 15$.

Решение. Разложим многочлен $x^{15} - 1$ над полем F, применяя Sage.

$$x^{15} - 1 = (x+1) * (x^2 + x + 1) * (x^4 + x + 1) * (x^4 + x^3 + 1) * (x^4 + x^3 + x^2 + x + 1).$$

Таким образом

$$FG \cong GF(2) \oplus GF(4) \oplus GF(16) \oplus GF(16) \oplus GF(16).$$

Упражнение 6.2. Найти описание групповой алгебры FG над полем $F = GF(3)$ циклической группы G порядка $k = 14$.

Упражнение 6.3. Найти описание групповой алгебры FG над полем $F = GF(5)$ циклической группы G порядка $k = 42$.

6.2 Лабораторная работа

В этой лабораторной работе предлагается исследовать строение групповой алгебры над группой диэдра.

Далее $F = GF(q)$ – конечное поле из q элементов. $G = D_n$ – группа диэдра порядка $2n$. Мы изучаем строение групповой алгебры в случае $(q, 2n) = 1$.

Нам понадобится следующее определение.

Определение 6.3. Для многочлена $f(x) \in F[x]$, такого что $f(0) \neq 0$ возвратным многочленом называется многочлен

$$f^*(x) = x^{deg(f)} f(\frac{1}{x}).$$

Если многочлены $f(x)$ и $f^*(x)$ имеют одинаковые корни в поле разложения, то многочлен f называется самовозвратным.

Пусть

$$x^n - 1 = [f_1 f_2 \dots f_r](f_{r+1})f^*_{r+1}) \dots (f_{r+s})f^*_{r+s})$$

разложение многочлена на произведение r самовозвратных и $2s$ несамовозвратных неприводимых многочленов, $f_1 = x - 1$. Если n – четное, то $f_2 = x + 1$.

Обозначим α_i – корень многочлена $f_i(x)$. Тогда α_i^{-1} являются корнями $f_i^*(x)$.

Строение алгебры $F_q D_{2n}$ описывается следующей теоремой ([8])

Теорема 6.4. $F = GF(q)$ – конечное поле из q элементов. $G = D_n$ – группа диэдра порядка $2n$, $(q, 2n) = 1$.

Тогда в случае n – нечетное

$$FG \cong F \oplus F \oplus \bigoplus_{j=2}^{r} M_2(F[\alpha_j + \alpha_j^{-1}]) \oplus \bigoplus_{j=r+1}^{r+s} M_2(F[\alpha_j]).$$

В случае n – четное

$$FG \cong F \oplus F \oplus F \oplus F \oplus \bigoplus_{j=3}^{r} M_2(F[\alpha_j + \alpha_j^{-1}]) \oplus \bigoplus_{j=r+1}^{r+s} M_2(F[\alpha_j]).$$

6.1. Найти разложение многочлена $x^3 - 1$ над $GF(7)$. Какие множители являются самовозвратными?

6.2. Найти разложение многочлена $x^3 - 1$ над $GF(5)$. Какие множители являются самовозвратными?

6.3. Найти разложение многочлена $x^6 - 1$ над $GF(7)$. Какие множители являются самовозвратными?

****

Далее рассматриваем вычисления по работе Карпова для $n = p^n$ (q, p -взаимно простые) и $n = 2p$

6.3 Темы дипломных работ:

1. Строение полупростых алгебр над группами диэдра получено в статьях [2], [3], [8] различными техниками. Исследовать другие классы конечных групп.

2. Рассмотреть групповые коды в случае модулярных групповых алгебр, то есть когда характеристика поля делить порядок группы.

3. Мы рассматривали групповые коды как двусторонние идеалы. Рассмотреть односторонние идеалы в качестве кодов. Исследовать какие результаты можно получить в этом случае, в

частности, можно ли получить коды с лучшими характеристиками. Начать можно с 'численных' экспериментов - вычислить характеристики в Sage и попробовать выдвинуть гипотезы.

6.3.1 Лабораторная работа

Эта лабораторная работа основана на статье Samir Assuena and César Polcino Milies *Good Codes From Dihedral Groups*, https://arxiv.org/abs/1506.03303, 2015

Рассмотрим группу диэдра

$$G = D_n =< a, b \mid a^n = 1, b^2 = 1, bab = a^{-1} >$$

и групповую алгебру FG над конечным полем F, состоящим из q элементов.

Нас будет интересовать частный случай: когда $n = p^m$, p – простое число, $(2p^m, q) = 1$.

Рассмотрим цепочку подгрупп

$$A = H_0 \supseteq H_1 \supseteq \ldots \supseteq H_m = \{1\},$$

где H_k подгруппа, порожденная a^{p^k}.

A также обозначает подгруппу, порожденную элементом a. Напомним, что подгруппа A – нормальная и G/A – циклическая группа порядка 2.

Пусть H – подгруппа группы G Обозначим через

$$\hat{H} = \frac{1}{|H|} \sum_{h \in H} h.$$

Упражнение 6.4. Покажите, что $\hat{H}$ – идемпотент.

Упражнение 6.5. Покажите, что если $B < C$ – подгруппы в G, то $\hat{C}\hat{B} = \hat{C}$.

Обозначим

$$e_0 = \hat{H}_0$$

$$e_k = \hat{H}_k - \hat{H}_{k-1} \quad (1 \leq k \leq m)$$

Справедлива следующая теорема (F. S. Dutra, R. A. Ferraz and C. Polcino Milies, Semisimple group codes and dihedral codes, Algebra and Disc. Math., 3 (2009), 28-48, Th 3.3)

Теорема 6.5. $\{\frac{1+b}{2}e_0, \frac{1-b}{2}e_0\} \cup \{e_k \mid 1 \leq k \leq m\}$ – образуют множество всех центральных примитивных идемпотентов.

Таким образом, мы можем перечислить все групповые коды. Напомним, что групповым годом мы назвали идеалы групповой алгебры.

Далее мы покажем (используя только вычисления в Sage), что односторонние идеалы могут дать коды с лучшими характеристиками.

Введем еще одно обозначение:

Пусть $e = e_k$ – один из идемпотентов $e_1, \ldots, e_m$. Обозначим

$$e_{11} = \left(\frac{1+b}{2}\right) e, \qquad e_{22} = \left(\frac{1-b}{2} e_0\right)$$

$$e_{12} = \left(\frac{1+b}{2}\right) a \left(\frac{1-b}{2}\right) e, \qquad e_{21} = 4[(a-a^{-1})e]^{-2} \left(\frac{1-b}{2}\right) a \left(\frac{1+b}{2}\right) e$$

$$f = e_{11} - e_{12}.$$

Справедлива следующая теорема (Prop. 2.5)

Теорема 6.6. Множество

$$\{f, af, a^2 f, \ldots a^{\varphi(p^k)-1}\}$$

является базисом (одностороннего) идеала FGf (здесь φ – функция Эйлера).

Пример.

Рассмотрим $G = D_9$, $F = GF(11)$.

Следующий код демонстрирует (не очень изящно, простым нахождением элементов с маленьким весом), что кодовые расстояния центральных кодов (то есть двусторонних идеалов) меньше 9

In [2]:
```
#Функция вычисления веса
def wt(x):
return len(str(x).split(sep='+'))
#Примитивные центральные идемпотенты
e0=H0
e11=(kG(1)+b)/R(2)*e0
e22=(kG(1)-b)/R(2)*e0
e1=H1-H0
e2=kG(1)-H1
#Некоторые веса элементов из минимальных двусторонних идеалов
print(wt(e11*(a**2)  - e11*a), wt(e22*(a**2)  - e22*a), wt(e1), wt(e2))
```

Out [2]: 1 1 9 3

Обозначим $e = e_1 = H_1 - H_0$, $f = e_{11} - e_{22}$ и односторонний идеал

$$FGf$$

.

В этом случае идеал FGf порожден двумя базисными элементами $\{f, af\}$.

Вычислим минимальный вес этого идеала над полем $GF(11)$, используя Sage

In [3]:
```
#Групповая алгебра над группой диэдра
G = DihedralGroup(9)
R=GF(11)
kG = G.algebra(R)
#Порождающие группы a,b
[a, b] = G.gens()
```

Out [3]:

Вычислим H_0 и H_1.

In [4]:
```
#Сумма всех элементов <a>
H0=kG(1)
for k in range(1,9):
    H0=H0+a**k
#Так мы показываем, что это элемент из кольца z=R(9)
H0=H0/R(9)

#Сумма всех элементов <a^3>
H1=kG(1)
for k in range(1,3):
    H1=H1+a**(3*k)
H1=H1/R(3)
```

Out [4]:

Вычисляем идемпотенты

In [5]:
```
E1=H1-H0
E11=(kG(1)+b)/R(2)*E1
E12=(kG(1)+b)/R(2)*a*(kG(1)-b)/R(2)*E1
f=E11-E12
```

Out [5]:

Перебираем все элементы идеала FGf и записываем их в список s.

In [6]:
```
s=[]
for i in range(9):
    for j in range(9):
        s.append(kG(i)*f+kG(j)*a*f)
```

Out [6]:

Выведем элементы s.

```
In [7]: s
```

```
Out [7]: [0, 4*() + 4*(2,9)(3,8)(4,7)(5,6) + 5*(1,2)(3,9)(4,8)(5,7) + 5*(1,2,3,4,5,6,7,8,9) + 2*(1,3)(4,9)(5
         + 2*(1,3,5,7,9,2,4,6,8) + 4*(1,4)(2,3)(5,9)(6,8) + 4*(1,4,7)(2,5,8)(3,6,9) + 5*(1,5)(2,4)(6,9)(7,8)
         + 5*(1,5,9,4,8,3,7,2,6) + 2*(1,6)(2,5)(3,4)(7,9) + 2*(1,6,2,7,3,8,4,9,5) + 4*(1,7)(2,6)(3,5)(8,9)
         + 4*(1,7,4)(2,8,5)(3,9,6) + 5*(1,8)(2,7)(3,6)(4,5) + 5*(1,8,6,4,2,9,7,5,3) + 2*(1,9,8,7,6,5,4,3,2)
         + 2*(1,9)(2,8)(3,7)(4,6), 8*() + 8*(2,9)(3,8)(4,7)(5,6) + 10*(1,2)(3,9)(4,8)(5,7) + 10*(1,2,3,4,5,
         + 4*(1,3)(4,9)(5,8)(6,7) + 4*(1,3,5,7,9,2,4,6,8) + 8*(1,4)(2,3)(5,9)(6,8) + 8*(1,4,7)(2,5,8)(3,6,9)
         + 10*(1,5)(2,4)(6,9)(7,8) + 10*(1,5,9,4,8,3,7,2,6) + 4*(1,6)(2,5)(3,4)(7,9) + 4*(1,6,2,7,3,8,4,9,5)
         + 8*(1,7)(2,6)(3,5)(8,9) + 8*(1,7,4)(2,8,5)(3,9,6) + 10*(1,8)(2,7)(3,6)(4,5) + 10*(1,8,6,4,2,9,7,5
         + 4*(1,9,8,7,6,5,4,3,2) + 4*(1,9)(2,8)(3,7)(4,6), () + (2,9)(3,8)(4,7)(5,6) + 4*(1,2)(3,9)(4,8)(5,7)
         + 4*(1,2,3,4,5,6,7,8,9) + 6*(1,3)(4,9)(5,8)(6,7) + 6*(1,3,5,7,9,2,4,6,8) + (1,4)(2,3)(5,9)(6,8)
         + (1,4,7)(2,5,8)(3,6,9) + 4*(1,5)(2,4)(6,9)(7,8) + 4*(1,5,9,4,8,3,7,2,6) + 6*(1,6)(2,5)(3,4)(7,9)
         + 6*(1,6,2,7,3,8,4,9,5) + (1,7)(2,6)(3,5)(8,9) + (1,7,4)(2,8,5)(3,9,6) + 4*(1,8)(2,7)(3,6)(4,5)
         + 4*(1,8,6,4,2,9,7,5,3) + 6*(1,9,8,7,6,5,4,3,2) + 6*(1,9)(2,8)(3,7)(4,6), 5*() + 5*(2,9)(3,8)(4,7)(
         + 9*(1,2)(3,9)(4,8)(5,7) + 9*(1,2,3,4,5,6,7,8,9) + 8*(1,3)(4,9)(5,8)(6,7) + 8*(1,3,5,7,9,2,4,6,8)
         + 5*(1,4)(2,3)(5,9)(6,8) + 5*(1,4,7)(2,5,8)(3,6,9) + 9*(1,5)(2,4)(6,9)(7,8) + 9*(1,5,9,4,8,3,7,2,6)
         + 8*(1,6)(2,5)(3,4)(7,9) + 8*(1,6,2,7,3,8,4,9,5) + 5*(1,7)(2,6)(3,5)(8,9) + 5*(1,7,4)(2,8,5)(3,9,6)
         + 9*(1,8)(2,7)(3,6)(4,5) + 9*(1,8,6,4,2,9,7,5,3) + 8*(1,9,8,7,6,5,4,3,2) + 8*(1,9)(2,8)(3,7)(4,6),
         9*() + 9*(2,9)(3,8)(4,7)(5,6) + 3*(1,2)(3,9)(4,8)(5,7) + 3*(1,2,3,4,5,6,7,8,9) + 10*(1,3)(4,9)(5,8
         + 10*(1,3,5,7,9,2,4,6,8) + 9*(1,4)(2,3)(5,9)(6,8) + 9*(1,4,7)(2,5,8)(3,6,9) + 3*(1,5)(2,4)(6,9)(7,8
         + 3*(1,5,9,4,8,3,7,2,6) + 10*(1,6)(2,5)(3,4)(7,9) + 10*(1,6,2,7,3,8,4,9,5) + 9*(1,7)(2,6)(3,5)(8,9)
         + 9*(1,7,4)(2,8,5)(3,9,6) + 3*(1,8)(2,7)(3,6)(4,5) + 3*(1,8,6,4,2,9,7,5,3) + 10*(1,9,8,7,6,5,4,3,2)
         + 10*(1,9)(2,8)(3,7)(4,6), ...
```

Такое представление мы получим из-за того, что группа диэдра в Sage реализована как подгруппа группы перестановок. Это позволяет Sage эффективно производить вычисления.

В следующем фрагменте мы подсчитаем веса

```
In [8]: l=[]
        for t in s:
            l.append(len(str(t).split(sep='+')))
        print(l)
```

```
Out [8]: [1, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 15, 15, 15, 18, 15, 15, 18, 18, 18, 15, 18, 15,
          15, 15, 15, 18, 15, 15, 18, 15, 15, 18, 18, 18, 18, 15, 18, 15, 18, 15, 15, 18, 18, 18, 15, 15,
          15, 15, 18, 18, 15, 18, 18, 18, 15, 18, 15, 18, 18, 15, 15, 18, 15, 15, 18, 18, 15, 15, 18, 15,
          18, 15, 15, 18, 18, 18, 15, 15, 18]
```

Вычислим наименьший вес

```
In [9]: from collections import Counter
        cnt = Counter(l).most_common(10)
        cnt
```

```
Out [9]: [(18, 44), (15, 36), (1, 1)]
```

Получили количество элементов различных весов.

Пара $(1, 1)$ здесь соответствует нулевому элементу. Такое представление обусловлено тем, как мы считаем веса. Нам пришлось представить элементы групповой алгебры как строки, а затем разбить их на отдельные элементы, используя в качестве разделителя знак '+'.

Таким образом, мы получили $[18, 2, 15]$-код.

Упражнение 6.6. Найдите минимальное расстояние в случае когда поле $GF(5), GF(7), GF(13)$.

Покажите, что если брать поле $GF(5)$ или $GF(7)$, то минимальное расстояние будет меньше 15.

Литература

[1] The Mathematical Theory of Coding - I. F. Blake and R. C. Mullin (New York: Academic Press, 1975, xi + 356 pp

[2] Raul Antonio Ferraz, César Polcino Milies *Idempotents in group algebras and minimal abelian codes* - Finite Fields and Their Applications, Volume 13, Issue 2, 2007, Pages 382-393, https://doi.org/10.1016/j.ffa.2005.09.007.

[3] Flaviana S. Dutra, Raul A. Ferraz and C. Polcino Milies *Semisimple group codes and dihedral codes* - Algebra and Discrete Mathematics, 2009., N 3. pp. 28 – 48

[4] Sarah Spence Adams *Introduction to Algebraic Coding Theory With Gap. Fall 2006*, https://pi.math.cornell.edu/ web3360/eccbook2007.pdf

[5] Абызов, А. Н. Кольца и модули : монография / А. Н. Абызов, А. А. Туганбаев. — Москва : ФЛИНТА, 2017. — 258 с. — ISBN 978-5-9765-2940-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/97108 (дата обращения: 24.07.2023). — Режим доступа: для авториз. пользователей.

[6] Berman, S.D. Semisimple cyclic and Abelian codes. II. Cybern Syst Anal 3, 17–23 (1967). https://doi.org/10.1007/BF01119999

[7] Berman, S.D. On the theory of group codes. Cybern Syst Anal 3, 25–31 (1967). https://doi.org/10.1007/BF01072842

[8] F.E. Brochero Martínez, Structure of finite dihedral group algebra, Finite Fields and Their Applications, Volume 35 (2015), 204-214, https://doi.org/10.1016/j.ffa.2015.05.002.