

Дискретная математика

Коды и информация

Константин Чухарев

Осень 2026

Расстояние Хэмминга

Информация — это снятие неопределённости.

— Клод Шеннон

Защита и сжатие

Канал передачи данных шумит. Биты могут переворачиваться.

Две задачи:

- **Защита:** добавить избыточность, чтобы исправить ошибки.
- **Сжатие:** убрать избыточность, чтобы уменьшить размер.

- Коды, исправляющие ошибки, — в QR-кодах, CD, памяти ECC.
- Сжатие без потерь — в zip, PNG.

Расстояние Хэмминга

Чтобы исправлять ошибки, нужно измерять расстояние между строками.

Определение 1 (Расстояние Хэмминга)

Число позиций, в которых строки различаются:

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

Пример

$d(10110, 10001) = 3$: различаются позиции 3, 4, 5.

$d(0000, 1111) = 4$, $d(x, x) = 0$.

Это метрика: симметрична, удовлетворяет неравенству треугольника.

Чтобы превратить x в y , нужно перевернуть $d(x, y)$ битов.

Кодовое расстояние

Определение 2

Кодовое расстояние (*minimum distance*) d — минимум расстояний между разными кодовыми словами.

Код с M словами длины n и расстоянием d — (n, M, d) -код.

Для линейных кодов (определение — в разделе «Коды и GF(2)») пишут (n, k, d) , где $k = \log_2 |C|$ — размерность кода.

Пример

Код Хэмминга $(7, 4, 3)$: $n = 7$, $2^4 = 16$ слов, $d = 3$.

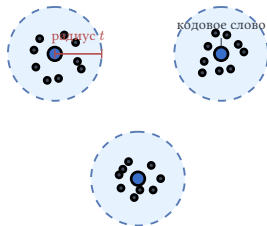
Код-повторение $(3, 1, 3)$: слова 000 и 111.

Расстояние и число единиц

Расстояние — число позиций, а не разность числа единиц.

10110 и 10001: единиц 3 и 2 — разность 1, но разошлись три позиции.

Исправление ошибок



Теорема 1 (Способность кода)

Если $d \geq s + 1$ — код **обнаруживает** до s ошибок, ведь слово с $\leq s$ перевёрнутыми битами отстоит от исходного меньше чем на d и не может совпасть с другим кодовым словом.

Если $d \geq 2t + 1$ — код **исправляет** до t ошибок.

Доказательство способности

Доказательство

Исправление: шары радиуса t вокруг разных кодовых слов не пересекаются. Иначе точка z на расстоянии $\leq t$ от двух центров x и y давала бы $d(x, y) \leq d(x, z) + d(z, y) \leq 2t$ — противоречие с $d \geq 2t + 1$. Значит слово с $\leq t$ ошибками попадает ровно в один шар и декодируется в его центр.

Шары радиуса t не пересекаются: принятое слово — в одном шаре, декодируем в центр.

Код Хэмминга (7, 4)

Параметры

Код Хэмминга — первый код, исправляющий ошибки.

Определение 3

Параметры: $n = 7$, $k = 4$ бита данных, $d = 3$.

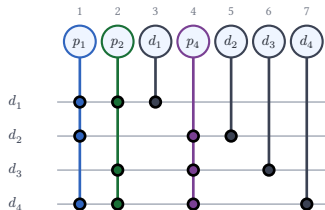
Позиции 1, 2 и 4 занимают проверочные биты, а позиции 3, 5, 6 и 7 — биты данных:

$$p_1, p_2, d_1, p_4, d_2, d_3, d_4.$$

Проверочный бит p_i — XOR данных своей группы.

Цель: XOR каждой группы равен нулю.

Контрольные группы



- p_1 контролирует d_1, d_2, d_4 ;
- p_2 контролирует d_1, d_3, d_4 ;
- p_4 — d_2, d_3, d_4 .

p_i контролирует позиции, где i -й бит номера равен 1. Позиция 7 — во всех группах.

Кодирование

Пример

Данные $d = (1, 0, 1, 1)$.

$$p_1 = 1 \oplus 0 \oplus 1 = 0.$$

$$p_2 = 1 \oplus 1 \oplus 1 = 1.$$

$$p_4 = 0 \oplus 1 \oplus 1 = 0.$$

Кодовое слово: $(0, 1, 1, 0, 0, 1, 1)$.

Кодирование и декодирование зеркальны: три XOR при кодировании и те же три контрольные группы при декодировании задают одну структуру.

Синдром и исправление

Определение 4 (Синдром)

При приёме вычисляем три бита, каждый — XOR своей группы.

Записанные как двоичное число $s_4s_2s_1$, три бита синдрома дают номер позиции с ошибкой, а нулевой синдром означает отсутствие ошибки.

Пример

Передано $(0, 1, 1, 0, 0, 1, 1)$, ошибка в позиции 7: принято $(0, 1, 1, 0, 0, 1, 0)$.

Синдром $(1, 1, 1) = 1 + 2 + 4 = 7$.

Переворачиваем бит 7 — ошибка исправлена.

Проверочная матрица

Синдром — «диагноз»: по симптомам распознаём болезнь.

Номер позиции совпал потому, что позиции 1..7 в двоичном виде — столбцы проверочной матрицы: синдром складывает столбцы искажённых позиций.

Проверочная матрица кода Хэмминга $(7, 4)$ — все ненулевые двоичные векторы длины 3 по столбцам:

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

Кодовые слова — решения $Hx^T = 0$ над $\text{GF}(2)$. Для слова $x = (0, 1, 1, 0, 0, 1, 1)$: $Hx^T = (0 + 0 + 1 + 1, 1 + 1 + 1 + 1, 0 + 1 + 0 + 1)^T = (0, 0, 0)^T$.

Две ошибки

Код с $d = 3$ исправляет одну ошибку.

Пример

Две ошибки в позициях 4 и 6.

Синдром указывает на неверную позицию — исправление затрагивает здоровый бит.

Две ошибки маскируются под одну.

Если ошибок больше, чем код гарантированно исправляет, он за результат не отвечает.

Расширенный код различает одиночную и двойную ошибку.

Линейные коды

Коды и GF(2)

Определение 5

Линейный код — множество кодовых слов, замкнутое относительно XOR.

Сумма двух кодовых слов — снова кодовое слово.

Пример

Сумма двух кодовых слов кода Хэмминга: $(0, 1, 1, 0, 0, 1, 1) \oplus (1, 1, 0, 1, 0, 1, 0) = (1, 0, 1, 1, 0, 0, 1)$ — снова кодовое слово.

Линейное подпространство

Все двоичные строки длины n — векторное пространство $\text{GF}(2)^n$, где $\text{GF}(2)$ — поле из двух элементов $\{0, 1\}$:

- XOR — сложение,
- AND — умножение,
- сложение векторов — побитовый XOR.

Линейный код — подпространство.

Код Хэмминга — линейный: его можно описать матрицей.

Расширенный код (8, 4, 4)

Один дополнительный бит чётности увеличивает расстояние до 4.

Теорема 2

Расширенный код по-прежнему исправляет одну ошибку.

Но теперь отличает двойную от одиночной.

Доказательство

Четыре комбинации «синдром / бит чётности» (ниже) однозначно различают:

- ошибок нет,
- испорчен бит чётности,
- одна ошибка,
- две ошибки.

Четыре исхода

Четыре исхода:

- Синдром 0, чётность сошлась — ошибок нет.
- Синдром 0, чётность нет — испорчен бит чётности.
- Синдром не 0, чётность нет — одна ошибка, исправляем.
- Синдром не 0, бит чётности совпал — две ошибки, не трогаем.

При подозрении на две ошибки декодер сигнализирует о повреждении и не вносит исправление.

Код Хаффмана

Сжатие без потерь

Код Хаффмана решает обратную задачу: убрать избыточность.

Частым символам — короткие кодовые слова, редким — длинные.

Как азбука Морзе: Е — точка, Q — длинная последовательность.

Определение 6 (Префиксный код)

Ни одно кодовое слово не служит префиксом другого.

Пример

Код $\{0, 10, 11\}$ префиксный: ни одно слово не начало другого. Код $\{0, 01, 011\}$ — нет: 01 служит префиксом 011.

Декодируется однозначно и без задержки.

Жадный алгоритм

Алгоритм строит дерево снизу вверх.

Пример

Частоты: $A : 0.40$, $B : 0.25$, $C : 0.20$, $D : 0.10$, $E : 0.05$.

Повторяем, пока не останется одно дерево: берём два наименьших веса и сливаем их в узел с суммарным весом.

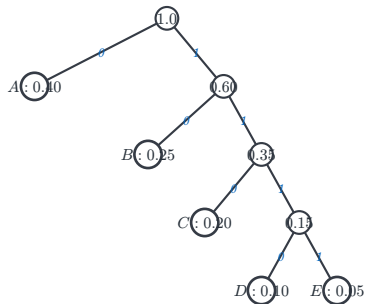
1. D и E (0.15);
2. C и DE (0.35);
3. B и CDE (0.60);
4. A и BCDE (1.00).

Коды: $A : 0$, $B : 10$, $C : 110$, $D : 1110$, $E : 1111$.

Оптимальность жадного алгоритма

Жадный выбор оптимален: если в оптимальном дереве более редкий символ сидит выше более частого, обмен местами не увеличивает среднюю длину, поэтому два наименьших веса всегда можно слить первыми.

Дерево Хаффмана



Путь от корня к листу — кодовое слово.

Средняя длина 2.10 против 3 у равномерного кода.

Пакетная передача: за пределами кодов

Коды исправляют искажённое значение, но предполагают, что символ дошёл. Для каналов, где пакет может потеряться, нужна доставка, а не алгебра.

Данные режутся на пакеты, пакеты идут по сети независимо. Порядок не гарантирован: пакет может потеряться, прийти дважды, переставиться. Помехоустойчивый код здесь бессилён — он не возвращает потерянную позицию.

UDP и TCP

Транспортный уровень решает доставку.

UDP — простейший транспорт: отправляет пакет и не проверяет, дошёл ли он. Скорость и простота, без гарантий.

TCP — поверх IP, восстанавливает надёжность. Порядковые номера, подтверждения (ACK), таймауты, повторная передача. Плата — задержка и повторные отправки.

Коды (CRC, Хэмминг, RS) защищают данные внутри пакета, TCP — доставку пакета. Разные уровни, не заменяют друг друга.

Что мы поняли?

- Расстояние Хэмминга — метрика на строках, а кодовое расстояние d даёт гарантию: $d \geq 2t + 1$ исправляет t ошибок.
- В коде Хэмминга проверочные биты — XOR контрольных групп, а синдром называет номер искажённой позиции.
- Линейный код — подпространство в $\text{GF}(2)^n$. Расширенный код отличает одиночную ошибку от двойной.
- Защита добавляет избыточность, сжатие убирает её: код Хаффмана даёт кратчайший префиксный код.

Вопросы для самопроверки

- Вычислите расстояние Хэмминга между 101101 и 110010.
- Почему $d = 3$ исправляет одну ошибку, но не две?
- Закодируйте данные $(1, 1, 0, 0)$ кодом Хэмминга $(7, 4)$.
- Принято слово $(1, 0, 1, 1, 0, 0, 0)$. Вычислите синдром и исправьте ошибку.
- Почему расширенный код отличает двойную ошибку от одиночной?
- Что такое префиксный код и зачем он нужен?
- Постройте код Хаффмана для частот $(0.5, 0.25, 0.125, 0.125)$.