# Теория кодирования

Михайлов Максим

23 сентября 2022 г.

# Оглавление

# Лекция 1

# 2 сентября

## 1 Введение

**Определение. Передаваемый сигнал** это

$$x(t) = \sum_i S_{x_i}(t - iT)$$

, где $x_i$ — передаваемые символы, $T$ — продолжительность передачи одного символа.

*Пример* ($M$-ичная амплитудно-импульсная модуляция).

$$S_i(t) = \alpha(2i + 1 - M)g(t)\sin(2\pi ft)$$

, где:

- $g(t)$ — сигнальный импульс
- $f$ — несущая частота
- $\alpha$ — коэффициент энергии передаваемого сигнала

Модели канала:

1. В непрерывном времени: $y(t) = x(t) + \eta(t)$
2. В дискретном времени: $y_i = \alpha(2x_i + 1 - M) + \eta_i$

$\eta$ — белый шум, обычно Гауссов $\mathcal{N}(0, \sigma^2)$.

У передаваемого сигнала обычно не должно быть постоянной компоненты (по причинам физики), поэтому сигнал симметричен. С точки зрения теории кодирования это несущественно.

Приемник наблюдает на выходе канала вектор $y = (y_0 \dots y_{n-1})$. Канал характеризуется условным распределением $P_{Y|X}(y \mid x)$, где $X, Y$ — случайные величины, соответствующие векторам переданных и принятых символов.

Приемник разбивает векторное пространство на решающие области $R_x : y \in R_x \implies \hat{x} = x$, т.е. если сигнал попал в область $R_x$, то мы ему сопоставляем кодовый символ $x$. Тогда вероятность ошибки:

$$\begin{aligned} P_e &= \int_{\mathbb{R}^N} P_e(y) p_Y(y) dy \\ &= \sum_x \int_{R_x} p_e(y) p_Y(y) dy \\ &= \sum_x \int_{R_x} (1 - p_{X|Y}\{x \mid y\}) p_Y(y) dy \\ &= 1 - \sum_x \int_{R_x} p_{X|Y}\{x \mid y\} p_Y(y) dy \end{aligned}$$

Приемник должен найти оптимальные решающие области. Оптимальность определяется критерием, например:

1. Критерий максимума апостериорной вероятности (критерий идеального наблюдателя)

$$\begin{aligned} R_x &= \{y \mid p_{X|Y}(x \mid y) > p_{X|Y}(x' \mid y), x' \neq x\} \\ &= \{y \mid P_X(x) p_{Y|X}(y \mid x) > P_X(x') p_{Y|X}(y \mid x'), x' \neq x\} \end{aligned}$$

2. Критерий максимума правдоподобия

$$R_x = \{y \mid p_{Y|X}(y \mid x) > p_{Y|X}(y \mid x'), x' \neq x\}$$

   В случае равновероятных символов этот критерий совпадает с критерием идеального наблюдателя.

*Пример* (2-ичная амплитудно-импульсная модуляция (2-АМ)). Пусть $y_i = \alpha(2x_i - 1) + \eta_i, \eta_i \sim \mathcal{N}(0, \sigma^2), x_i \in \{0, 1\}$. Тогда:

$$p_{Y|X}(y \mid x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y - \alpha(2x-1))^2}{2\sigma^2}}$$

Применим критерий максимального правдоподобия:

$$R_0 = \{y \mid y < 0\}, R_1 = \{y \mid y \geq 0\}$$

Вычислим вероятность ошибки:

$$P_e = P_X(0) P\{Y \geq 0 \mid X = 0\} + P_X(1) P\{Y < 0 \mid X = 1\}$$

$$
\begin{aligned}
&= 0.5 \int_0^\infty p_{Y|X}(y \mid 0) dy + 0.5 \int_{-\infty}^0 p_{Y|X}(y \mid 1) dy \\
&= \int_0^\infty \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y+\alpha)^2}{2\sigma^2}} dy \\
&= \int_\alpha^\infty \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{y^2}{2\sigma^2}} dy \\
&= \int_{\frac{\alpha}{\sigma}}^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy \\
&=: Q\left(\frac{\alpha}{\sigma}\right) \\
&= \frac{1}{2}\operatorname{erfc}\left(\frac{\alpha}{\sqrt{2}\sigma}\right)
\end{aligned}
$$

Значение сигнала это обычно уровень напряжения. Как мы знаем из школьной физики, мощность $P = \frac{U^2}{R}$. Мы хотим минимизировать мощность, чтобы экономить электроэнергию. Мощность сигнала суть случайная величина с матожиданием, пропорциональным $E_S = \alpha^2$. Мощность белого шума не зависит от частоты и пропорциональна $\sigma^2 = \frac{N_0}{2}$. Если же шум зависит от частоты, то он называется розовым или голубым.

Соотношение мощностей сигнал/шум на символ это $\frac{E_S}{N_0}$, обычно измеряемое в децибелах, т.е. $10 \log_{10} \frac{E_S}{N_0}$. Однако нас интересуют не символы, а биты и тогда соотношение сигнал/шум на бит это $\frac{E_S}{RN_0}$, где $R$ — количество бит информации, представленных одним символом.

## 1.1 Способы снижения вероятности ошибки

1. Посимвольное повторение: будем передавать вместо каждого символа $m$ копий того же символа.

   $$y_{mi+j} = \alpha(2x_i - 1) + \eta_{mi+j}, \quad 0 \le j < m$$

   Рассмотрим разные способы работы приемника:

   - Для каждого $y_{mi+j}$ проведем голосование. Тогда вероятность ошибки:

     $$P_v(m) = \sum_{j=\lceil \frac{m}{2} \rceil}^{m-1} C_m^j P_e^j (1 - P_e)^{m-j}$$

   - Примем решение по

     $$\sum_{j=0}^{m-1} y_{mi+j} = m\alpha(2x_i - 1) + \sum_{j=0}^{m-1} \eta_{mi+j}$$

, т.е сложим наблюдения в одном блоке. Тогда вероятность ошибки:

$$P_a(m) = Q\left(\frac{m\alpha}{\sqrt{m}\sigma}\right) = Q\left(\sqrt{2\frac{mE_s}{N_0}}\right) = Q\left(\sqrt{2\frac{E_b}{N_0}}\right)$$

Выигрыша не будет, т.к. на один символ передается в $m$ раз меньше бит (см. последний переход).

Второй метод лучше первого, т.к. мы не теряем информацию о нашей уверенности в каждом принятом символе.

Избыточность на уровне битов не дала улучшения, поэтому введем избыточность на уровне блоков.

## 1.2 Понятие кода

**Определение. Код** — множество допустимых последовательностей символов алфавита $X$.

Последовательности могут быть конечными или бесконечными, но на практике только конечными. Не каждая последовательность символов алфавита является кодовой.

**Определение. Кодер** — устройство, отображающее информационные последовательности символов алфавита $\mathcal{B}$ в кодовые.

Различным последовательностям алфавита $\mathcal{B}$ сопоставляются различные последовательности алфавита $X$ для однозначности кодирования.

**Определение. Скорость кода** — отношение длин информационной и кодовой последовательностей.

**Определение. Декодер** — устройство, восстанавливающее по принятой последовательности символов наиболее вероятную кодовую последовательность.

## 1.3 Теоремы кодирования

Пусть для передачи используется код $\mathcal{C} \subset X^n$ длины $n$, состоящий из $M$ кодовых слов, выбираемых с одинаковой вероятностью.

**Теорема 1** (обратная)**.** Для дискретного постоянного канала с пропускной способностью $C$ для любого $\delta > 0$ существует $\varepsilon > 0$ такое, что для любого кода со скоростью $R > C + \delta$ средняя вероятность ошибки $\overline{P_e} \geq \varepsilon$

**Теорема 2** (прямая)**.** Для дискретного постоянного канала с пропускной способностью $C$ для любых $\varepsilon, \delta > 0$ существует достаточно большое число $n_0 > 0$, такое что для всех натуральных $n \geq n_0$ существует код длиной $n$ со скоростью $R \geq C - \delta$, средняя вероятность ошибки которого $\overline{P_e} \leq \varepsilon$

## 1.4 Пропускная способность некоторых каналов

1. Двоичный симметричный канал: $X, Y \in \{0, 1\}, p_{Y|X}(y \mid x) = \begin{cases} p, & y \neq x \\ 1-p, & y = x \end{cases}$

$$C_{BSC} = 1 + p \log_2 p + (1-p) \log_2 (1-p)$$

2. Идеальный частотно ограниченный гауссовский канал: $y(t) = x(t) + \eta(t), \eta(t)$ — гауссовский случайный процесс, спектральная плотность мощности которого равна $S(f) = \begin{cases} \frac{N_0}{2}, & -W < f < W \\ 0, & \text{иначе} \end{cases}$ Из теории случайных процессов:

$$C_{AWGN} = W \log_2 \left(1 + \frac{E_S}{W N_0}\right)$$

$$\lim_{W \to \infty} C_{AWGN} = \frac{E_S}{N_0 \ln 2}$$

# Лекция 2

# 9 сентября

Не дописано

# Лекция 3

# 16 сентября

**Определение.** Для группы $\mathcal{G} = (G, +)$ с подгруппой $\mathcal{H} = (H, +)$ назовём **смежным классом** для $x \in G$ множество:

$$x + H = \{x + h \mid h \in H\}$$

*Пример.* Для группы $\mathbb{Z}$ группа чётных чисел $2\mathbb{Z}$ является подгруппой. Для 1 смежный класс — все нечётные числа, для 2 — все чётные.

Не дописано

Иногда демодулятор может сообщить, что некоторый полученный символ ненадежен. Это называется **стиранием**. Также стиранием считаются потери пакетов в компьютерных сетях (UDP).

Стирания исправлять проще, чем ошибки, т.к. мы знаем, где они произошли.

**Определение. Весовой спектр кода** это $A_i = |\{c \in C \mid \mathrm{wt}(c) = i\}|$

Для двоичного симметричного канала с переходной вероятностью $p$ вероятность необнаружения ошибки это:

$$P\{S = 0\} = \sum_{i=d}^{n} A_i p^i (1-p)^{n-i} \leq \sum_{i=d}^{n} C_n^i p^i (1-p)^{n-1}$$

Не дописано

Для аддитивного гауссовского канала с двоичной модуляцией вероятность ошибки мягкого декодирования по максимуму правдоподобия линейного блокового кода:

$$P \leq \sum_{i=d}^{n} A_i Q\left(\sqrt{2i\frac{E_s}{N_0}}\right) = \sum_{i=d}^{n} A_i Q\left(\sqrt{2iR\frac{E_b}{N_0}}\right) = \frac{1}{2}\sum_{i=d}^{n} A_i \mathrm{erfc}\left(\sqrt{iR\frac{E_s}{N_0}}\right)$$

**Определение. Полное декодирование по минимальному расстоянию** — нахождение по $y$ ближайшего кодового слова $c = \text{argmin}_{c \in C} d(c, y)$

**Определение. Информационная совокупность *(ИС)*** — множество из $k$ позиций в кодовом слове, значения которых однозначно определяют значения символов на остальных позициях кодового слова.

**Определение.** Если $\gamma = \{j_1 \dots j_k\}$ — информационная совокупность, то остальные позиции кодового слова называются **проверочной совокупностью**.

Если $\gamma$ — информационная совокупность, то матрица из столбцов порождающей матрицы с номерами $j_1 \dots j_k$, обратима. Почему? Не дописаноБудем обозначать такую матрицу $M(\gamma)$.

*Примечание.* Эта матрица не единственна.

Пусть $G(\gamma) = M(\gamma)G$ — порождающая матрица, содержащая единичную подматрицу на столбцах $j_1 \dots j_k$.

Информационные совокупности позволяют более эффективно проверять коды, т.к. стандартный метод требует таблицу размера $2^k$.

**Теорема 3.** Алгоритм декодирования по информационной совокупности обеспечивает полное декодирование по минимальному расстоянию.

*Доказательство.* Необходимо доказать, что для всякого исправимого вектора ошибки $r$ существует информационная совокупность, свободная от ошибок.

Пусть $c$ — единственное решение задачи декодирования по минимальному расстоянию. Тогда $e = r - c$ — вектор ошибки, $E = \text{supp}(e)$ — множество позиций ненулевых элементов $e$ и $|E| \leq n - k$.

Пусть $N = \{1 \dots n\}$. Предположим, что $N \setminus E$ не содержит информационных совокупностей, тогда существуют различные кодовые слова, отличающиеся от $r$ в позициях $E$. Но тогда $c$ не единственно, противоречие. □

Сложность декодирования для $(n, k)_q$ кода экспоненциальная.

## 1.5 Покрытия

**Определение. Покрытием** $M(n, m, t)$ называется такой набор $F \subset 2^{N_n}$, где $N_n = \{1 \dots n\}, \forall f \in F \ \ |f| = m$ и любое $t$-элементное подмножество $N_n$ содержится в одном из $f \in F$.

Для декодирования по информационной совокупности с исправлением не более $t$ ошибок необходимо покрыть все исправимые конфигурации ошибок. Элементы такого покрытия — проверочные совокупности.

*Пример.* Не дописано

Построение минимального покрытия — NP-полная задача. Но есть итеративный приблизительный алгоритм.

## 1.6 Дуальные коды

Не дописано

## 1.7 Граница Хэмминга

**Теорема 4.** Для любого $q$-ичного кода с минимальным расстоянием $d = 2t + 1$ число кодовых слов удовлетворяет

$$A_q(n, d) \leq \frac{q^n}{\sum_{i=0}^{t} C_n^i (q-1)^i}$$

*Доказательство.* Если код способен исправить $t$ ошибок, то вокруг всех кодовых слов можно описать хэмминговы шары радиуса $t$, не пересекающиеся друг с другом. □

### 1.7.1 Асимптотическая оценка

$$\begin{aligned} A_q(n, d) &\leq \frac{q^n}{\sum_{i=0}^{t} C_n^i (q-1)^i} \\ &\leq \frac{q^n}{C_n^{\frac{d-1}{2}} (q-1)^{\frac{d-1}{2}}} \\ &= \frac{q^n \left(\frac{d-1}{2}\right)! \left(n - \frac{d-1}{2}\right)!}{n! (q-1)^{\frac{d-1}{2}}} \\ &= ??? \end{aligned}$$

## 1.8 Граница Варшамова–Гилберта

**Теорема 5.** Существует $q$-ичный код длины $n$ с минимальным расстоянием $d$, число слов которого удовлетворяет:

$$A_q(n, d) \geq \frac{q^n}{\sum_{i=0}^{d-1} C_n^i (q-1)^i}$$

*Доказательство.* Если код $C$ имеет максимальную мощность, для любого вектора $x \notin C$ существует кодовое слово $c$ такое, что $d(x, c) \leq d - 1$. Не дописано □

## 1.9 Граница Варшамова–Гилберта для линейных кодов

**Теорема 6.** Если

$$q^r > \sum_{i=0}^{d-2} C_{n-1}^i (q-1)^i$$

, то существует линейный код над $GF(q)$ длины $n$ с минимальным расстоянием не менее $d$ и не более чем $r = n - k$ проверочными символами.

*Доказательство.* Построим матрицу $H$ размера $(n-k) \times n$ такую, что любые $d-1$ её столбцов линейно независимы.

Первый столбец пусть будет произвольным ненулевым вектором. Если уже выбраны $j$ столбцов, то $j+1$-й столбец не может быть никакой линейной комбинацией любых $d-2$ выбранных столбцов. Таких столбцов $\sum_{i=0}^{d-2} C_j^i (q-1)^i$. Пока не запрещены все $q^{n-k}$ столбцов, то можно выбрать ещё хотя бы один столбец. □

Не дописано

## 1.10 Граница Грайсмера

*Обозначение.* $N(k,d)$ — минимальная длина двоичного линейного кода размерности $k$ с минимальным расстоянием $d$.

**Теорема 7.** $N(k,d) \geq d + N(k-1, \lceil d/2 \rceil)$

*Доказательство.* Пусть порождающая матрица $(n,k,d)$ кода $C$ наименьшей длины $n = N(k,d)$ имеет вид:

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

Не дописано □

# Лекция 4

# 23 сентября

## 2 Универсальные методы декодирования линейных кодов

Мы изучали методы жесткого декодирования, но на практике чаще применяются методы мягкого декодирования, которые учитывают надежность символов.

Мы уже знаем, что декодирование кода по критерию максимального правдоподобия в канале с АБГШ эквивалентно декодированию по критерию минимального расстояния Евклида.

Несложными преобразованиями получим:

???

Пусть тогда $\hat{c}_i = \begin{cases} 0, & y_i > 0 \\ 1, & y_i \leq 0 \end{cases}$ — жесткие решения.

Тогда:

$$\text{argmax}_{c \in C} \sum_{i=0}^{n-1} (-1)^{c_i} y_i = \text{argmax}_{c \in C} \sum_{i=0}^{n-1} (-1)^{c_i} y_i$$
$$= ???$$

Не дописано

## 2.1 Метод порядковых статистик

Рассмотрим передачу кодовых слов $c_0 \dots c_{n-1}$ двоичного $(n, k)$ кода с помощью символов 2-АМ, например $y_i = (-1)^{c_i} + \eta_i$ — с АБГШ.

Пусть тогда $L_i = \log \frac{P\{c_i=0|y_i\}}{P\{c_i=1|y_i\}}$ — логарифмические отношения правдоподобия. Вероятность ошибки в $i$-том жестком решении убывает с увеличением $|L_i|$. Тогда выберем информационную совокупность $J$ для кода, соответствующего наибольшим значениям $|L_i|$. Приведем порождающую матрицу кода к виду $G_J$ с единичной подматрицей в столбцах $J$.

С большой вероятностью число неверных решений $\hat{c}_j$, где $j \in J$, мало. Переберем все конфигурации ошибок $e$ веса не более $t$ на $J$ и построим кодовые слова $c_e = (\hat{c}_J + e)G_J$. Из всех полученных кодовых слов выберем наиболее правдоподобное.

Сложность алгоритма $\mathcal{O}\left(k^2 n + \sum_{i=0}^{t} \text{in} C_k^i\right)$.

Есть два[1] способа ускорить этот алгоритм:

1. Обменять память на скорость каким-то образом (придумать дома).
2. Реализовать раннюю остановку.

Рассмотрим двоичный $(n, k, d)$ код $C$ с порождающей матрицей $G$, пусть $D(x, y)$ — функция расстояния Хемминга.

Существует много кодов, содержащих одинаковые префиксы некоторой длины $a$, поэтому можно не пересчитывать $\sum_{i=0}^{a} D(c_i, y_i)$ заново.

Не дописано

## 2.2 Декодирование по решеткам

**Определение. Решетка** ***(англ. ???)*** — граф, обладающий следующими свойствами:

1. Вершины графа разбиты на непересекающиеся подмножества, называемые **уровнями** или **ярусами**.
2. Нулевой и последний ярусы содержат по одной вершине, называемой **терминальной**.
3. Граф направленный и допускается движение только от уровня с меньшим номером к уровню с большим номером.
4. Ребрам графа сопоставлены метки, соответствующие символам кодовых слов, а также метрики, называемые весами.

На таком графе можно запустить алгоритм Дейкстры или алгоритм Витерби

[1] Не взаимозаменяющих

**Определение. Профиль сложности решетки** это $\xi_0 \dots \xi_n$, где $\xi_i = |V_i|$.

**Определение.** Решетка называется **минимальной**, если профиль сложности решетки минимален среди всех решеток с заданным количеством ярусов.

Рассмотрим все кодовые слова $c_m = c_{m,0} \dots c_{m,n-1}$ кода.

Для любого $i$ определим префикс длины $i$, называемый **прошлым** и обозначим его $c_m^p$, а оставшийся суффикс длины $n - i$ — **будущим** и обозначим его $c_m^f$.

Очевидно, что в произвольной решетке пути, входящие в фиксированную вершину, имеют общее будущее, а пути, исходящие из фиксированной вершины, имеют общее прошлое.

Не дописано

Докажем, что построенная решетка минимальна.

*Доказательство.* Рассмотрим произвольную решетку $T'$ этого кода.

В $T'$ два слова $c_1 = (c_1^p, c_1^f)$ и $c_2 = (c_2^p, c_2^f)$ могут иметь общую вершину на ярусе $i$ только если $F_i(c_1^p) = F_i(c_2^p)$. По построению два пути, проходящие через общую вершину в $T'$, проходят также через общую вершину в $T$.

Таким образом, число вершин на ярусе $i$ в $T'$ не меньше числа вершин на ярусе $i$ в $T$. □

**Теорема 8.** Любой код имеет минимальную решетку, и все минимальные решетки совпадают с точностью до нумерации вершин яруса.

Не дописано

**Теорема 9.** Решетка, получаемая по порождающей матрице в минимальной спановой форме, минимальна.

*Доказательство.* Докажем, что для любого $l \in \mathbb{N}$ пути, определяющие слова с одинаковыми $c^f$ длины $n - l$, не проходят через различные узлы на ярусе с номером $l$.

Узел, через который проходит путь на ярусе $l$, определяется значениями информационных символов, которые соответствуют активным на этом ярусе строкам. Т.к. эти строки линейно независимы и заканчиваются на ярусах с номерами $> l$, следовательно, нетривиальные линейные комбинации этих строк отличаются хотя бы на одной позиции $> l$.

Предположим, что есть два слова с одинаковым будущим, проходящие через разные узлы на ярусе $l$. Их сумма образует слово, активное на ярусе $l$ и равное $0$ на позициях правее $l$. Но из соображений выше таких слов быть не может, а следовательно слова с одинаковым будущим проходят через одни и те же узлы, а следовательно решетка минимальна. □

Не дописано

**Теорема 10.** Решетка, построенная по проверочной матрице, минимальна.

*Доказательство.* Докажем, что пути с одинаковыми $c^f$ не проходят через разные узлы.

Для кодового слова $c = (c^p, c^f)$ частичные синдромы, вычисленные по $c^p$ и $c^f$, совпадают. Следовательно, все совпадающие $c^f$ исходят из одного и того же узла, определенного частичным синдромом $c^p$. □

# 3 Декодирование с мягким выходом

Длинные коды можно строить путём комбинирования более коротких кодов. Как — мы узнаем позже. Декодеры таких кодов могут быть построены из декодеров кодов-компонент. Такие декодеры могут взаимодействовать путем обмена апостериорными вероятностями:

$$p\{c_i = a \mid y_0^{n-1}\} = \sum_{c \in C_i(a)} p\{c \mid y_0^{n-1}\}$$

???

## 3.1 Алгоритм Бала–Коке–Елинека–Равива

Нужно вычислить

$$L_i = \ln \frac{P\{c_i = 0 \mid y_0^{n-1}\}}{P\{c_i = 1 \mid y_0^{n-1}\}} = \ln \frac{\sum_{(s',s)\in S_0} \dfrac{p(s_i = s', s_{i+1}, y_0^{n-1})}{p(y_0^{n-1})}}{\sum_{(s',s)\in S_0} \dfrac{p(s_i = s', s_{i+1}, y_0^{n-1})}{p(y_0^{n-1})}}$$

, где $S_0$ и $S_1$ — множества пар состояний $s' \in V_i, s \in V_{i+1}$, переход между которыми помечен 0 и 1 соответственно, $p(y_0^{n-1})$ — совместная плотность распределения принятых сигналов, $p(s_i = s', s_{i+1} = s, y_0^{n-1})$ — совместная плотность распределения принятых сигналов и состояний кодера на ярусах $i$ и $i+1$.

Поведение кодера при обработке $i$-го информационного бита определяется только его состоянием $s'$ на предыдущем шаге и канал не имеет памяти:

$$p(s_i = s', s_{i+1} = s, y_0^{n-1}) = ???$$