Лекции по Теории Кодирования 7 семестр

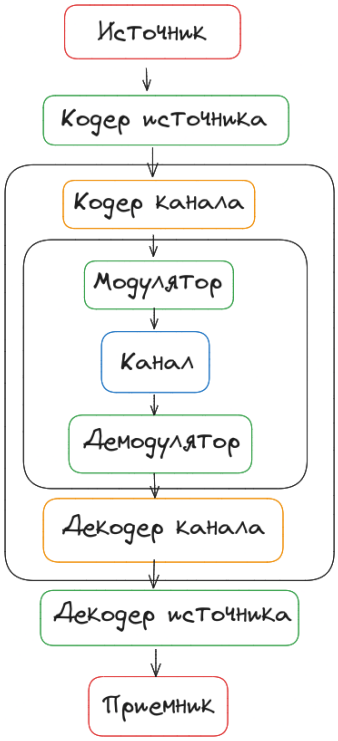
Ilya Yaroshevskiy

25 февраля 2024 г.

Оглавление

Лекция 1

1.1 Введение



1.1.1 Модулятор

Определение. Передаваемый сигнал равен

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

, где x_i – передаваемые символы, T – продолжительность символьного интервала

Пример. M-ичная амплитудно-импульсная модуляция

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

, где $g(t)$ – сигнальный импульс (например, единичный импульс продолжительностью T), f – несущая частота, α – коэффицент, определяющий энергию передаваемого сигнала

Пример. Модель канала в непрервном времени $y(t) = x(t) + \eta(t)$

Пример. Модель канала в дискретном времени $y_i = (2x_i + 1 - M) + \eta_i$

Определение. $\eta_i \sim \mathcal{N}(0, \sigma^2)$ – канал с **аддитивным белым гауссовским шумом**

1.1.2 Приемник

Замечание. Приемник наблюдает на выходе канала вектор $y = \begin{pmatrix} y_0 & \dots & y_{n-1} \end{pmatrix}$.
Канал характеризуется условным распределением $p_{Y|X}(y|x)$, где X, Y – случайные величины, соответствующие векторам переданных и принятых символов. Если выход канала – непрерывная случайная величина, $p_{Y|X}(y|x)$ – условная плотность вероятности. Приемник реализует некоторое разбиение векторного пространства на решающие области $R_x : y \in R_x \implies \hat{x} = x$

Определение. Вероятность ошибки

$$P_e = \int_{\mathbb{R}^N} p_e(y) p_Y(y) dy = \sum_x \int_{R_x} p_e(y) p_Y(y) dy =$$

$$= \sum_x \int_{R_x} (1 - p_{X|Y}(x|y)) p_Y(y) dy = 1 - \sum_x \int_{R_x} p_{X|Y}(x|y) p_Y(y) dy$$

Хотим минимизировать P_e:

Определение. Критерий максимума апостериорной вероятности (**критерий идеального наблюдателя**)

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

Определение. Критерий максимума правдоподобия

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

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:

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

Применим критерий максимального правдоподобия:

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

Вычислим вероятность ошибки:

$$P_e = P_X(0) P\{Y \geq 0 | X = 0\} + P_X(1) P\{Y < 0 | X = 1\} = \dots = \int_{\frac{\alpha}{\sigma}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}} dy = Q\left(\frac{\alpha}{\sigma}\right) =$$

$$= \frac{1}{2} erfc\left(\frac{\alpha}{\sqrt{2}\sigma}\right)$$

Замечание. Значение сигнала это обычно уровень напряжения. Как мы знаем 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.

1.2 Понятие кода

Определение. Код – множество допустимых последовательностей символов алфавита X, как конечных так и бесконечных

Замечание. На практике ограничиваются послеловательностями длины n

Замечание. Не всякая последовательность символов из X является кодовой

Определение. Кодер – устройство, реализующее отображение информационных последовательностей символов алфавита B в кодовые

Замечание. Различным информационным последовательностям сопоставляются разсличные кодовые последовательности

Определение. Скорость кода – отношение длин информационной и кодовой последовательностей

Определение. Декодер – устройство, восстанавливающее по принятой последовательности символов *наиболее вероятную* соответствующую ей кодовую (или информационную) последовательность

Замечание. Под наиболее вероятным подразумевается критерии иделального наблюдателя и максимального правдоподобия

1.2.1 Теорема кодирование

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

Теорема 1.2.1 (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 $\bar{P}_\varepsilon > \varepsilon$

Замечание. Постоянный канал – статистические свойства со временем не меняются
Дискретный канал – вход и выход дискретные

Замечание. Здесь говориться о том канал характеризуется величиной C. Если попробуем передать данные с большей пропускной способностью, то вероятность ошибки будет ограничена снизу.

Теорема 1.2.2 (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 $P_\varepsilon \leq \varepsilon$

1.2.2 Пропускные способности каналов

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

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

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

$$C_{\text{AWGN}} = W\log_2\left(1 + \frac{E_s}{WN_0}\right)$$

1.2.3 Мягкое и жесткое декодирование

Канал с аддитивным белым гауссовским шумом: $y_i = (2x_i - 1) + \eta_i, x_i \in \{0,1\}$

Определение. Мягкое декодирование: декодер непосредственно использует y_i

Определение. Жесткое декодирование: декодер использует оценки $\hat{x}_i$

1.2.4 Спектральная эффективность

Определение. Спектральная эффективность кодирования $\beta = \frac{R}{W}$ $\left[\frac{\text{бит}}{\text{с Гц}}\right]$

Лекция 2

2.1 Декодирование

Определение. Критерий минимального расстояния $X = Y$. Декодер ищет кодово слово

$$c = \text{argmin}_{c \in \mathcal{C}} d(c, y)$$

Определение. Алгоритм называется алгоритмом полного декодирования по критерию K, если он способен найти решение соответствующуй оптимизационной задачи для любого возможного принятого сигнала

2.2 Метрики

Определение. Функция $d(x, y)$ называется метрикой, если:

- $d(x, y) \geq 0, d(x, y) = 0 \Leftrightarrow x = y$
- $d(x, y) = d(y, x)$
- Неравенстно треугольника $d(x, y) + d(y, z) \geq d(x, z)$

Определение. Метрическое пространство – множество X с определенной на нем метрикой

Пример. Расстояние Хемминга $d_H(x, y) = |\{i | x_i \neq y_i\}|$. Двоичный симметричный канал ($p < 0.5, X = Y = \{0, 1\}$:

$$\hat{c} = \text{argmax}_{x \in \emptyset} \prod_{i=1}^{n} P\{y_i | c_i\} = \cdots = \text{argmin}_{c \in \mathcal{C}} \sum_{i=1}^{n} a|y_i - c_i|$$

Пример. Расстояние Евклида $d_E(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$. Аддитивный Гауссовский канал с амплитудно-импульсной модуляцией ($Y = \mathbb{R}^n$).

Пример. Расстояние Ли ($A = GF(q)^n$) : $d_L(x, y) = \sum_{i=1} \min(|x_i - y_i|, q - |x_i - y_i|)$. Аддитивный Гауссовский канал с q-ичной фазовой модуляцией

Пример. Ранговое расстояние ($A = GF(q)^{n \times m}$) : $d_R(x, y) = \text{rank}(x - y)$. Сетевые коды

2.3 Кодирование

Операция канального кодирования вносит в информационную последовательность избыточность, необходимую для последующего исправления возможных ошибок

Замечание. Блоковые коды преобразуют блок из k символов в блок из n символов. Преобразование отдельных блоков выполняется независимо

Замечание. Сверточные коды преобразуют блок из k символов в блок из n символов. Преобразование зависит от предыдущих блоков.

2.3.1 Блоковые коды

Замечание. n – длина кода $\mathcal{C}$. Для исправления ошибок требуется, чтобы не все $|X|^n$ последовательностей были кодовыми словами. Мощность кода (число различных кодовых слов) $M = |C|$.

Замечание. Скорость кода: $R = \frac{\log_{|X|} M}{n}$.

Определение. Минимальным расстоянием кода называется минимальное расстояние Хемминга между его различными кодовыми словами

Пример. Пример $C = \{000, 111\}, d_{m\times n}(C) = 3$

Замечание. Хеммингов шар радиуса $d_{\min -1}$, описанный вокруг кодового слова $c \in C$, не содержит никаких других кодовых слов

Утверждение. *Код с минимальныи расстоянием Хемминга d способен исправить* $\left\lfloor \frac{d-1}{2} \right\rfloor$

2.3.2 Линейный код

Определение. Линейным (n, k) **(длины** n **размерностью** k**) кодом** C называется k-мерное линейное подпространство n-мерного линейного пространства над полем $GF(q)$.

Замечание. Число кодовых слов равно q^k

Определение. Порождающая $k \times n$ матрица полного ранга $G : C = \{y = xG | x \in GF(q)^k\}$

Определение. Проверочная матрица $r \times n$, $H : C = \{y \in GF(q)^n | yH^T = 0\}$, $r \geq n - k = \text{rank}(H)$

$$GH^T = 0$$

Замечание. С помощью линейных операций над строками и перестановок столбцов порождающая матрица может быть приведена к виду $G = (I|A)$.
NB Вообще говоря перестановкой столбцов получается порождающая матрица другого кода, поэтому перестановка столбцов в порождающей матрице должна быть согласована с перестановкой в проверочной

Замечание. Систематическое кодирование $xG = (x|xA)$ – информационный вектор является подвектором кодого слова. Применение систематического кодирования упрощает декодирование

$$H = (A^T| - I)$$

Утверждение. *Минимальное расстояние линейного блокового кода $\mathcal{C}$ равно* $d = \min\limits_{\substack{c' \neq c'' \\ c', c'' \in \mathcal{C}}} d(c', c'') = \min_{c \in \mathcal{C} \setminus \{0\}} wt(c)$*, где $wt(c)$ – вес вектора (количество единиц)*

Доказательство. Расстояние между двумя векторами – число позиций в которых они отличаются

$$d(c', c'') = \sum_{i=0}^{n} d(x, y) = \sum_{i=0}^{n} d(0, x - y) = d(0, c' - c'')$$

как код образует линейное подпространство, значит он группа по сложение, значит $c' - c''$ – кодовое слово. $wt(c)$ – вес Хемминга □

Утверждение. *Если H – проверочная матица кода длины n, то код имеет размерность $n - r \Leftrightarrow$ существуют r линейно независимых столбцов матрицы H, а любые $r + 1$ столбцов линейно зависимы*

Утверждение. *Если H – проверочная матрица кода длины n, то код имеет минимальное расстояние $d \Leftrightarrow$ любые $1, 2, \ldots, d - 1$ столбцов H линейно независимы, но существуют d линейно зависимых столбцов матрицы H*

Замечание. Принадлежность коду $yH^T = 0$ эквивалентна ЛЗ столбцов

Утверждение. *Граница Синглтона (верхняя): для любого (n, k, d) линейного кода $n - k \geq d - 1$*

Следствие 2.3.0.1. Ранг матрицы H (максимальное число ЛНЗ столбцов) не может превосходить $n - k$

Определение. Граница Синглтона для произовльных кодов $A_q(n, d) \leq q^{n-d+1}$

Определение. Коды с $n - k = d - 1$ называются **разделимыми** кодами с максимальным достижимым расстоянием

2.3.3 Простейшие коды

Пример. Пусть G – обратимая $n \times n$ матрица. Она порождает код $(n, n, 1)$.

Пример. $(n, 1, n)$ код с n-кратным повторением: $G' = (11 \ldots 1)$,

$$H' = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 & 1 \\ 0 & 1 & 0 & \ldots & 0 & 1 \\ 0 & 0 & 1 & \ldots & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & 1 \end{pmatrix}$$

Пример. $(n, n - 1, 2)$ код с проверкой на четность:

$$G'' = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 1 \\ 0 & 1 & 0 & \dots & 0 & 1 \\ 0 & 0 & 1 & \dots & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 1 \end{pmatrix}, H'' = \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix}$$

2.4 Декодирование

2.4.1 Код Хемминга

Определение. Выберем в качестве столбцов матрицы H все ненулевые двоичные векторы длины r:

- Длина кода $n = 2^r - 1$
- Размерность $k = n - r = 2^r - r - 1$
- Минимальное расстояние $d = 3$

Пример. Если столбцы выписаны в соответсвии с двоичным кодом:

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

2.4.2 Жесткое и мягкое декодирование

В классичесической архитектуре предполагается что модулятор преобразует закодированные данные в сигнал, демодулятор оценивает символы кодовых слов, а декодер потом пытается исправить ошибки. Такой подход плох тем что теряется информация о надежности отдельных принятых символов. Сейчас как правило используют мягкое декодирование: демодулятор каким-то образом формирует информацию о надежности отдельных принятых символов, декодер при исправлении пытается учесть эту информацию.

2.4.3 Жесткое декодирование линейных кодов

Замечание. Рассмотрим двоичный симметричный канал с переходной вероятностью $p < 0.5$. Предположим что передатчик использует линейный блоковый код с порождающей матрицей G. Тогда принятый вектор $y = xG + e$, где e – вектор ошибки, который содержит 1 на тех позициях где произошла ошибка.

Определение. Синдром принятого вектора $S = yH^T = xGH^T + eH^T = eH^T$ зависит только от вектора ошибки

Определение. Пусть есть подгруппа некоторой группы G, $G' \subset G$. Возьмем элемент $a \in G$, то **смежным классом** подгруппы G', называется $aG' = \{a \cdot x | x \in G'\} \subseteq G$

Определение. Лидер смежного класса – минимальный по весу вектор.

Замечание. Рассмотрим все возможные вектора e и выпишем соответсвующие синдромы. Отсортируем по весу все возможные вектора e, соответствующие каждому возможному значению синдрома (*стандартная расстановка*). В качестве решения задачи декодирования выбираем самый легкий вектор e, соответствующий вычисленному синдрому:

2.4.4 Код хемминга (продолжение)

Замечание. Если произошла только одна ошибка, то $S = eH^T$ будет равно какому-то столбцу матрицы H. Получается синдром – двоичное представление числа, которое является номером позиции в которой произошла ошибка.

2.4.5 Стирание

Замечание. Некоторые символы могут просто теряться. Стирания могут происходить одновременно с ошибками. Утверждается что (n, k, d) код может исправить любую комбинацию из t ошибок и v стираний, если $d \geq 2t + v + 1$. Стирание эквивалентно выкалыванию кода на v позиций $\implies$ минимальное расстояние уменьшается не более чем на v.

Замечание. Декодирование ошибок и стираний для кодов над $GF(2)$:

- Положить все стертые позиции равными 0, исправить ошибки в полученном векторе
- Положить все стертые позиции равными 1, исправить ошибки в полученном векторе
- Выбрать результат декодирования, ближайший к принятому вектору

2.5 Качество (performace) декодирования

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

Замечание. Рассмотрим двоичный симметричный канал с переходной вероятностью p. Вероятность не обнаружения ошибки:

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

Вероятность правильного декодирования. Вероятность того, что вектор ошибки является лидером смежного класса:

$$P_{\text{correct}} = \sum_{i=0}^{l} L_i p^i (1-p)^{n-i}$$

, где L_i – число лидеров смежных классов веса i, l – максимальный вес лидера смежного класса

2.6 Декодирование по информационным совокупностям

Определение. Информационной совокупностью называется множество из k позиций в кодовом слове, значения которых однозначно определяют значения на остальных позициях кодового слова

Определение. Если $\gamma = \{j_1, \dots, j_k\}$ – ИС, то все прочие позиции $\{1, \dots, n\} \setminus \gamma$ образуют **проверочную совокупносить**

Утверждение. *Если $\gamma = \{j_1, \dots, j_k\}$ образует ИС, то матрица, составненная из столбцов $j_1, \dots, j_k$ порождающей матрицы, обратима*

Доказательство. $G = (A|B)$ – порождающая матрица. Если матрица A – необратима, у нее есть линейно зависимые столбцы или строки, значит $\exists x \neq 0 : xA = 0$. Есть кодовое $c = (c'|c'')$:

$$c + xG = (c'|c'') + (xA|xB) = (c' + xA|c'' + xB) = (c'|c'' + xB)$$

$xB \neq 0$ т.к. вся линейно зависимость осталась в матрице A. Получилось новое кодовое слово, которые совпадает с начальным на позициях c'. Получается что смотря на эти позиции нельзя однозначно указать значения на остальных позициях. Значит это не информационная совокупность. Противоречие, значит A – обратимая □

Определение. $M(\gamma) = A^{-1}$

Утверждение. $G(\gamma) = M(\gamma)G$ – *породжающая матрица, содержащая единичную подматрицу на столбцах γ, где $M(\gamma)$ – подходящая обратимая матрица*

Замечание. $G = (A|B), G(\gamma) = (\overbrace{I}^{\gamma}|M(\gamma)B)$

Определение. ИС **свободна от ошибок**, если соответствующие позиции вектора e равны 0: $e(\gamma) = 0$

2.6.1 Декодирование

Замечание. Декодирование $y = xG + e$ по информационным совокупностям:

- (первоначальный кандидат) $c = 0$
- Выбрать ИС γ. Вычислить $c' = y(\gamma)G(\gamma)$
- Если $d(c', y) < d(c, y), c = c'$
- Перейти к следующей ИС. Если все ИС проверены, возвратить c.
- Не всякие k позиций образуют информационную совокупность

Лекция 3

3.1 Декодирование по ИС

Теорема 3.1.1. Алгоритм декодирования по ИС обеспечивает полное декодирование по минимальному расстоянию

Доказательство. Необходимо доказать, что для всякого исправимого вектора ошибки существует информационная совокупность, свободная от ошибок Пусть c – единственное решение некоторой задачи декодирования по минимальному расстоянию

- $e = y - c$ – вектор ошибки
- $E = \text{supp}(e)$ – множество позиций ненулевых элементов e. $|E| \leq n - k$

Пусть $N = \{1, 2, \ldots, n\}$. Предположим, что $N \setminus E$ не содержит информационных совокупностей $\Longrightarrow$ существует различные кодовые слова, отличающиеся от принятого вектора в позициях E. Это противоречит предположению о единственности c. □

3.1.1 Покрытия

Определение. $M(n, m, t)$ **прокрытием** называется такой набор $F \subset 2^{N_n}$ из подмножеств мощности m множества $N_n = \{1, 2, \ldots, n\}$, что всякое t-элементное подмножество N_n содержится в одном из $f \in F$

Декодирование на ИС с исправлением не более t ошибок: необходимо покрыть все исправимые конфигурации ошибок. Элементы покрытия задают проверочные совокупности.

Пример. Пример декодирования $(7, 4, 3)$ кода, порождаемого $G = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$

- $y = (0, 1, 1, 0, 1, 0, 0)$
- Все возможные конфигурации единичных ошибок покрываются проверочными совокупностями $M(7, 3, 1) = \{\{1, 2, 4\}, \{5, 6, 7\}, \{3, 4, 5\}\}$
- Им соответствуют информационные совокупности $\{\{3, 5, 6, 7\}, \{1, 2, 3, 4\}, \{1, 2, 6, 7\}\}$
- Преобразованные порождающие матрицы

$$\begin{pmatrix}1&1&1&0&0&0&0\\1&0&0&1&1&0&0\\0&1&0&1&0&1&0\\1&1&0&1&0&0&1\end{pmatrix}\sim\begin{pmatrix}1&0&0&0&0&1&1\\0&1&0&0&1&0&1\\0&0&1&0&1&1&0\\0&0&0&1&1&1&1\end{pmatrix}\sim\begin{pmatrix}1&0&0&1&1&0&0\\0&1&1&1&1&0&0\\0&0&1&0&1&1&0\\0&0&1&1&0&0&1\end{pmatrix}$$

- Кодовые слова, соответсвующие подвекторам принятого вектора:

$$\underline{(0,1,1,1,1,0,0)},(0,1,1,0,0,1,1),\underline{(0,1,1,1,1,0,0)}$$

3.2 Дуальные коды

Определение. Пусть задан (n,k) код с проверочной матрицей H. **Дуальным** к нему называется $(n,n-k)$ код с порождающей матрицей H.

Определение. Кодовые слова дуального кода – множество всех проверок на четность исходного кода

Замечание. Скалярное произведение кодового слова из дуального кода на слово из исходного кода равно 0.

Определение. Самодуальным называется код, совпадающий со своим дуальным

Утверждение. *Код с проверочной матрицей $H=(A|I)$ самодуален ттогда, когда A – квадратичная матрица, такая что $AA^T=-I$*

$$HH^T=AA^T+I$$

3.3 Весовой спектр кода

Определение. Спектром линейного кода называется последовательность $A_i, i=0\dots n$, где A_i равно числу кодовых слов веса i

Определение. Весовая функция кода

$$W(x,y)=\sum_{i=0}^{n}A_i x^{n-i}y^i=\sum_{u\in C}x^{n-wt(u)}y^{wt(u)}$$

Теорема 3.3.1 (Мак-Вильямс для двоичных линейных кодов)**.** Весовая функция кода C связана с весовой функцией дуального к нему кода $C_\perp$ соотношением

$$W_{C_\perp}=\frac{1}{|C|}W_C(x+y,x-y)$$

3.4 Границы

3.4.1 Граница Хемминга

Теорема 3.4.1 (Граница Хемминга)**.** 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 исправлять t ошибок, то вокруг всех кодовых слов можно описать хемминговы шары радиуса t, не пересекающиейся друг с другом

При $n \to \infty$ скорость q-ичного кода удовлетворяет $R \leq 1 - h_q\left(\frac{d}{2n}\right)$, где

$$h_q(x) = -x \log_q x - (1-x) \log_q (1-x)$$

Аппроксимация Стирлинга $n! \approx 2^{n \log_2 n + o(1)}$

$$A_q(n, d) \leq \frac{q^n}{\sum_{i=0}^{t} C_n^i (q-1)^i} \leq \frac{q^n}{C_n^{\lfloor (d-1)/2 \rfloor} (q-1)^{(d-1)/2}} = \frac{q^n |(d-1)/2|! (n - |(d-1)/2|)!}{n! (q-1)^{(d-1)/2}} \approx$$

$$\approx 2^{n \log_2 q + \frac{d-1}{2} \log_2 \left(\frac{d-1}{2}\right) + \left(n - \frac{d-1}{2}\right) \log_2 \left(n - \frac{d-1}{2}\right) - n \log_2 n - \frac{d-1}{2} \log_2 (q-1)}$$

$$R = \frac{\log_q A_q(n, d)}{n} \leq \log_q 2 (\log_2 q + \delta \log_2 \delta + (1-\delta) \log_2 (1-\delta) + \frac{\delta}{2} \log_2 (-1)), \delta = \frac{d-1}{n} \approx \frac{d}{n}$$

□

3.4.2 Граница Варшамова-Гильберта

Теорема 3.4.2 (Граница Варшамова-Гильберта)**.** Существует q-ичный код длины n с минимальными 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 $c : d(x, c) \leq d - 1$

Итеративное построение кода:

1. $A \coloneqq GF(q)^n$ (q^n различных векоторов)
2. Выберем произвольный вектор $c \in A$ и добавим его в код
3. Удалим из A шар радиуса $d-1$ с центром в c. Число элементов в шаре $\sum_{i=0}^{d-1} C_n^i (q-1)^i$. Число удаляемых элементов может быть меньше, т.к. некоторая часть шара могла быть удалена ранее
4. Если A не пусто, перейти к п. 2

□

3.4.3 Граница Варшамова-Гильберта для линейных кодов

Теорема 3.4.3 (Граница Варшамова-Гильберта для линейных кодов)**.** Если выполняется $q^r > \sum_{i=0}^{d-2} C_{n-1}^i (q-1)^i$, то существует линейный код над $GF(q)$ длины n с минимальными расстоянием не менее d и не более чем $r = n - k$ проверочными символами

Доказательство. Построим $(n-k) \times n$ матрицу H, такую что любые ее $d-1$ столбцов ЛНЗ

- Первый столбец – произвольный ненулевой вектор
- Если уже выбраны j столбцов, в качестве $(j+1)$-го не могут использоваться никакие линейные комбинации любых $d-2$ выбранных столбцов, число которых равно $\sum_{i=0}^{d-2} C_j^i (q-1)^i$
- Если запрещены еще не все q^{n-k} векторов, можно выбрат еще один столбец

□

Замечание. Существует (n,k,d) код над $GF(q)$, где $A_q(n,d) \geq q^k$, где k – наибольшее целое, такое что $q^k < \frac{q^n}{\sum_{i=0}^{d-2} C_{n-1}^i (q-1)^i}$ Граница ВГ для произвольных кодов: $A_q(n,d) \geq \frac{q^n}{\sum_{i=0}^{d-1} C_n^i (q-1)^i}$. Существует бесконечная последовательность двоичных линейных кодов со скоростью $R \leq 1 - g(d/n)$.

3.4.4 Граница Грайсмера

Определение. $N(k,d)$ – минимальная длина двоичного линейного кода размерности k с минимальными расстоянием d.

Теорема 3.4.4 (Граница Грайсмера)**.** $N(k,d) \geq d + N(k-1, \lceil d/w \rceil)$

Доказательство. Будем считать, что порождающая матрица (n,k,d) кода C наименьшей длины $n = N(k,d)$ имеет вид

$$G = \Big(\underbrace{\begin{matrix} 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ & & & G' & & & \end{matrix}}_{N(k,d)-d} \Big| \underbrace{\begin{matrix} 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ & & & * & & & \end{matrix}}_{d} \Big)$$

G' порождает код $(n-d, k-1, d')$. Пусть $(u|v) \in C,\ wt(u) = d' \implies d' + wt(v) \geq d$

$(u|1-v) \in C \implies d' + d - wt(v) \geq d \implies 2d' \geq d \implies d' \geq \lceil d/w \rceil \implies N(k-1, \lceil d/w \rceil) \leq N(k,d) - d$

□

Замечание.

$$N(k,d) \geq d + N(k-1, \lceil d/2 \rceil) \geq d + \lceil d/2 \rceil + N(k-2, \lceil d/4 \rceil) \geq \dots \geq \sum_{i=0}^{k-1} \left\lceil \frac{d}{2^i} \right\rceil$$

Теорема 3.4.5 (Граница Мак-Элиса-Родемича-Рамсея-Велча)**.**

$$R \leq h\left(\frac{1}{2} - \sqrt{\frac{d}{n}\left(1 - \frac{d}{n}\right)} \right)$$

3.5 Другие формулировки задачи мягкого декодирования

3.5.1 Критерии мягкого декодирования

Утверждение. *Декодирования кода C по критерию максимуму правдоподобия в канале с АБГШ эквивалентно декодированию по критерию минимального расстояния Евклида*

Замечание. Рассмотрим передачу по каналу с АБГШ символов $(-1)^{c_i}, c_i \in \{0,1\}$, т.е. $y_i = (-1)^{c_i} + \eta_i$

$$\text{argmin}_{c\in C} \sum_{i=0}^{n-1}(y_i - (-1)^{c_i})^2 = \text{argmin}_{c\in C} \sum_{i=0}^{n-1}(y_i^2 - 2(-1)^{c_i}y_i + (-1)^{2c_i}) = \text{argmax}_{c\in C} \sum_{i=0}^{n-1}(-1)^{c_i}y_i$$

Пусть $\hat{c_i} = \begin{cases} 0 & , y_i > 0 \\ 1 & , y_i \le 0 \end{cases}$ – жесткие решения

$$\text{argmax}_{c\in C} \sum_{i=0}^{n-1}(-1)^{c_i}y_i = \text{argmax}_{c\in C} \sum_{i=0}^{n-1}((-1)^{c_i}y_i - (-1)^{\hat{c}_i}y_i) = \text{argmax}_{c\in C} \sum_{i:c_i\neq\hat{c}_i} -|y_i| = \text{argmin}_{c\in C} E(c,y)$$

, где $E(c,y) = \sum_{i:c_i\neq\hat{c}_i} |y_i|$ – *корреляционная невязка.* y_i может быть заменено на логарифмические отношения правдоподобия $L_i = \log \frac{P\{c_i=0|y_i\}}{P\{c_i=1|y_i\}}$

Замечание. Задача минимизации евклидового расстояния эквивалентна задаче максимизации корреляция и эквивалентная задаче минимизации корреляционной невязки

3.5.2 Метод порядковых статистик

Замечание. Рассмотрим передачу кодовых слов $(c_0, \ldots, c_{n-1})$ двоичного (n,k) кода с помощью символов 2-АМ по каналу без памяти. Пусть $(y_0, \ldots, y_{n-1})$ – соответствующие принятые символы.

Пример. $y_i = (-1)^{c_i} + \eta_i,\ \eta_i \sim N(0, \sigma^2)$

- Пусть $L_i = \log \frac{P\{c_i=0|y_i\}}{P\{c_i=1|y_i\}}$ – логарифмические отношения правдоподобия
- Пусть $\hat{c}_i = \begin{cases} 0 & , L_i > 0 \\ 1 & , L_i < 0 \end{cases}$ – жесткие решения

Вероятность ошибки в $\hat{c}_i$ убывает с увеличением $|L_i|$. Выберем информационную совокупность J кода, соответствующую наибольшим значениям $|L_i|$. Приведем порождающую матрицу кода к виду G_J с единичной подматрицей в столбцах J. С большой вероятностью число неверных жестких решений $\hat{c}_i, i \in J$, мало. Переберем все конфигурации ошибок e веса не более t на J и построим кодовые слова $c_e = (\hat{c}_J + e)G_J$. Выберем наиболее правдоподобное из полученных кодовых слов. Сложность $O(k^2 n + \sum_{i=0}^{t} inC_k^i)$. При $t = d/4$ достигается вероятность ошибки, близкая к вероятности ошибки декодирования по максимуму правдоподобия

Лекция 4

4.1 Решетки

4.1.1 Мотивация

Замечание. Рассмотрим двоичный (n, k, d) код C с порождающей матрицей G. Пусть $D(x, y)$ – функция расстояния Хемминга. Декодирвание векотора y по максимуму правдоподобия

$$\hat{c} = \text{argmin}_{c \in C} D(c, y) = \text{argmin}_{c \in C} \sum_{i=0}^{n-1} D(c_i, y_i)$$

Существует много кодовых слов содержащих одинаковые префиксы $(c_0, \ldots, c_a)$. Было бы неразумно пересчитывать $\sum_{i=0}^{a} D(c_i, y_i)$ для них каждый раз заново. Декодирование – задача поиска ближайшего кодового слова. Попытаемся сформулировать ее как задачу поиска кратчайшего пути на графе

4.1.2 Решетка

Определение. Решеткой (trellis) называется граф, обладающий следующими свойствами

- Вершины разбиты на непересекающиеся подмножества (уровни или ярусы)
- Нулевой и последний ярусы содержат по 1 узду (терминальные узлы)
- Грав направленный. Допускается движение только от уровня с меньшим номером к уровню с болшим нжомером. Стрелки при этом, как правило, не рисуют
- Ребрам графа приписаны метки, соответсвующие символам кодовых слов, а также метрики, называемые также весами или длинами. Длина пути равна сумме длин входящих в него ребер. Пример метрики ребра на ярусе i, помеченного c: $D(c, y_i)$

Замечание. Сопоставим пути в решетке между терминальными узлами кодовым словам. Тогда задача поиска кодового слова, минимизирующего некоторую аддитивную функцию функцию длины эквивалентна задаче поиска кратчайшего пути между терминальными узлами решетки

4.1.3 Алгоритм Витерби

Program 1 Viterbi(y)

```
1: M_{0,0} = 0
2: for i = 1, ..., n do
3:    for v ∈ V_i do
4:       M_{v',v} = M_{i-1,v'} + D(c[i, v', v], y_{i-1}) {Для каждого входящего ребра (v', v) вычислить метрику его пути}
5:       M_{i,v} = min M_{v',v} {Метрика узла}
6:       B_{i,v} = argmin M_{v',v} {Узел-предшественник}
7:    end for
8: end for
9: v = 0
10: for i = n, ..., 1 do
11:    ĉ_{i-1} = c[B_{i,v}, v]
12:    v = B_{i,v}
13: end for
14: return ĉ, M_{n,0}
```

Замечание.

- $M_{i,v}$ – метрика узла v на ярусе i
- V_i – множества узлов на ярусе i
- $c[i, v', v]$ – метка ребра между узлами $v' \in V_{i-1}, v \in V_i$
- Число сложений не превосходит E (число ребер в решетке)
- Число сравнений не превосходит $E - V$, где V – число узлов

4.1.4 Минимальная решетка кода

Определение. Профиль сложности решетки: $(\xi_0, \ldots, \xi_n), \xi_i = |V_i|$ – число узлов на i-м ярусе

Определение. Решетка называется **минимальной**, если профиль сложности $(\xi'_0, \ldots, \xi'_n)$ любой другой решетки удовлетворяет $\xi'_i \geq \xi_i, 0 \leq i \leq n$.

Замечание. Как построить минимальную решетку

Выпишем все кодовые слова $e_m = (c_{m,0}, \ldots, c_{m,n-1})$ рассмтриваемого кода. Для некоторого i определим прошлое $c^p_m = (c_{m,0}, \ldots, c_{m,i-1})$ и будущее $c^f_m = (c_{m,i}, \ldots, c_{m,n-1})$. В любой решетке пути, входящие в некоторый узел, имеют общее будущее, а пути, исходящие из одного узла имеют общее прошлое.

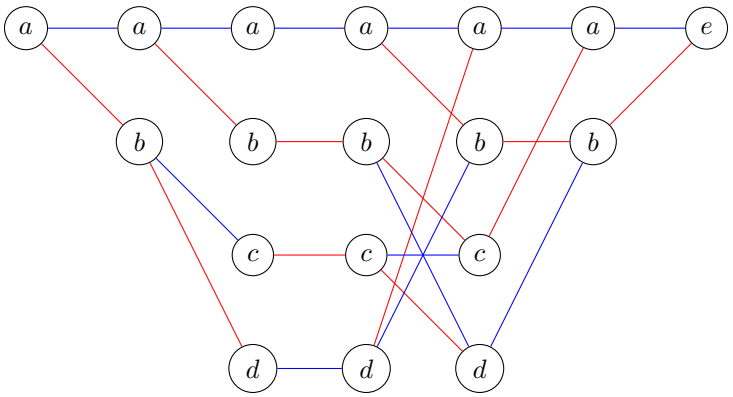
$F_i = \{c^f | \exists c^p : (c^p, c^f) \in C\}$ может быть единственным образом разбито на подмножество $F_i(c^p) = \{c^f | (c^p, c^f) \in C\}$. Сопоставим каждому такому подмножеству узлы на ярусе i. Узел $v \in V_i$ свяжем с узлом $v' \in V_{i+1}$, если для некоторого кодового слова прошлое, соответствующее v' является продолжением на 1 символ одной из последовательностей, ведущих в узле v. Пометим этим символом ребро (v, v').

Пример. Порождающая матрица

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

Кодовые слова: $c_0 = (000000), c_1 = (110100), c_2 = (101010), c_3 = (011110), c_4 = (101101), c_5 = (011001), c_6 = (000111), c_7 = (110011)$

i	c^p	$F_i(c^p)$	v_i	v_{i-1}	$c_{i,v_i,v_{i-1}}$
0	$\varnothing$	$\{c_m \mid 0 \le m \le 7\}$	a	–	–
1	0	$00000, 11110, 11001, 00111$	a	a	0
	1	$10100, 01010, 01101, 10011$	b	a	1
2	00	$0000, 0111$	a	a	0
	01	$1110, 1001$	b	a	1
	10	$1010, 1101$	c	b	0
	11	$0100, 0011$	d	b	1
3	000	$000, 111$	a	a	0
	011	$110, 001$	b	b	1
	101	$010, 101$	c	c	1
	110	$100, 011$	d	d	0
4	$0000, 1101$	00	a	a, d	$0, 1$
	$0001, 1100$	11	b	a, d	$1, 0$
	$0111, 1010$	10	c	b, c	$1, 0$
	$0110, 1011$	01	d	a, c	$0, 1$
5	$00000, 11010, 10101, 01111$	0	a	a, c	$0, 1$
	$10110, 01100, 00011, 11001$	1	b	b, d	$1, 0$
6	$\{c_m \mid 0 \le m \le 7\}$	$\varnothing$	a	a, b	$0, 1$



синие ребра – 0, красные – 1

Утверждение. *Полученная решетка T минимальна*

Доказательство. Рассмотрим произвольную решетку T' этого кода. В T' два слова $c_1 = (c_1^p, c_1^f), c_2 = (c_2^p, c_2^f)$ могут иметь общую вершину на ярусе i только если $F_i(c_1^p) = F_i(c_2^p)).,, T'$, проходят также через общий узел в T. Обратное не верно. Следовательно, число узлов на ярусе i в T' не меньше числа узлов на этом же ярусе в T. □

Теорема 4.1.1. Всякий код имеет минимальную решетку, все минимальные решетки совпадают с точностью до нумерации узлов каждого яруса

4.1.5 Минимальная спэновая форма матрицы

Замечание. Для (n, k) линейного кода C для каждого $i, 0 \leq i \leq n$ прошлое и будущее также являются линейными кодами. Для простроения решетки удобно найти базисы (порождающие матрицы) этих кодов. Удобно сделать это сразу для всех i. Началом $b(x)$ вектора $(x_0, \ldots, x_{n-1})$ будем называть номер первого его ненулевого элемента. Концом $e(x)$ вектора $(x_0, \ldots, x_{n-1})$ будем называть номер последнего его ненулевого элемента. Элементы на позициях $b(x), \ldots, e(x) - 1$ будем называть активными.

Определение. Порождающая матрица называется **приведенной к минимальной спэновой форме** (МСФ), если все начала строк различны и все концы строк различны.

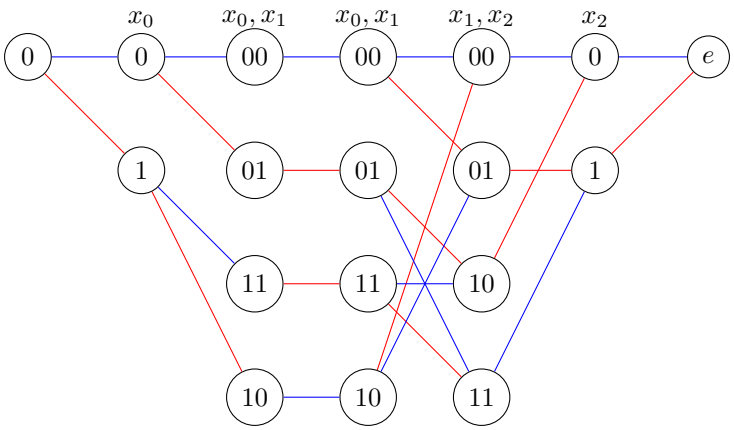
Замечание. Для определенности потребуем также, чтобы строки матрицы были упорядочены по возрастанию начал строк. Матрица может быть приведена к МСФ с помощью элементарных операций над строками

Пример. Приведем порождающую матрицу к МСФ

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{pmatrix} \xrightarrow{(1)+(2),(1)+(3),(2)\leftrightarrow(3)} \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

$$\xrightarrow{(2)+(1),(3)+(2)} \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

На ярусе i узлы нумеруются значениями информационных символов, соответсвующих строкам МСФ, активным в позиции i. Ребра помечаются линейными комбинациями активных элементов столбца порождающей матрицы. Полученная решетка минимальна



Теорема 4.1.2. Решетка, получаемая по порождающей матрице в МСФ, минимальна

Доказательство. Докажем, что $\forall l$ пути, определяющу кодовые 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, соответсвующих активным на этом ярусе строкам. Эти строки ЛНЗ и заканчиваются на ярусах с номерами $> l \implies$ нетривиальные линейные комбинации этих строк отличаются хотя бы на одной позиции правее l. Предположим, что есть 2 слова с одинаковым будущим, проходящие через ранзые узлы на ярусе l. Их сумма образует слово, активное на ярусе l, равное 0 на позициях правее l. Таких слов быть не может, т.е слова с одинаковым будущим проходят через одни и те же узлы, т.е. решетка минимальна. □

4.1.6 Построение решетки по проверочной матрице

Замечание.

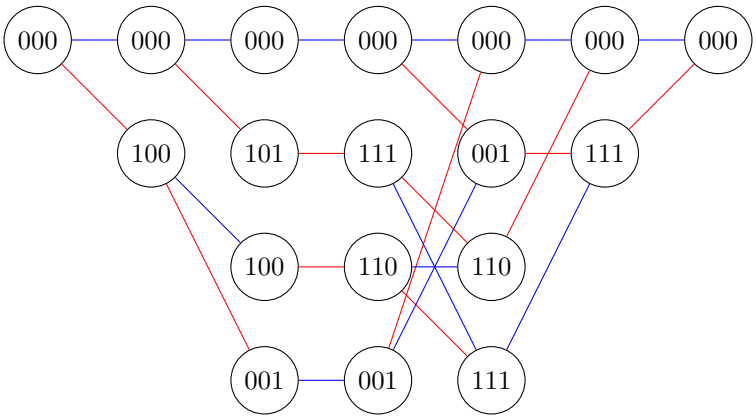
- Пусть дана проверочная матрица $H = (h_0, \dots, h_{n-1})$.
- Пусть $S_0 = 0$
- Накопленный синдром $S(x_0, \dots, x_{j-1}) = \sum_{i=0}^{j-1} h_i x_i$

Будем нумеровать узлы v в решетке накопленными синдромами $S(v)^T$. Существует ребро, помеченное c, из v' на ярусе i в v на ярусе $i+1$, если $S(v) = S(v') + ch_i$. Оставим единственный конечный узел, соответсвующий нулевому синдрому (т.е. допустимым кодовым словам). Удалим нетерминальные узлы, из которых не выходят ребра. Полученная решетка минимальна.

Замечание. Для реализации удобно (но не необходимо) привести проверочную матрицу к МСФ. Это упростит нумерацию узлов

Пример.

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



Теорема 4.1.3. 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, определяемого частичным синдромом c^p □

4.2 Декодирование с мягким выходом

Замечание. Длинные коды можно построить, комбинируя короткие. Декодеры "составных"кодов могут быть построены из декодеров компонентных кодов. Взаимодействие декодеров может осуществляться путем обмена апостериорными вероятностями

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

- $y_0^{n-1} = (y_0, \ldots, y_{n-1})$ – результат передачи кодовго слова кода C по каналу без памяти
- $C_i(a) = \{(c_0, \ldots, c_{n-1}) \in C | c_i = a\}$

Апостериорные логарифмические отношения правдоподобия для двоичных кодов

$$L_i = \ln \frac{p\{c_i = 0|y_0^{n-1}\}}{p\{c_i = 1|y_0^{n-1}\}}$$

Исходные данные ЛОПП символов кодового слова $L_i = \ln \frac{p(y_i|c_i=0)}{p(y_i|c_i=1)}$

Замечание. Такое декодирование называется декодированием с мягким входом и мягким выходом (soft input soft output, SISO)

4.2.1 Алгоритм Бала-Коке-Елинека-Равина

$$L_i = \ln \frac{P\{c_i = 0|y_0^{n-1}\}}{P\{c_i = 1|y_0^{n-1}\}} = \ln \frac{\sum_{(s',s)\in S_0} \frac{p(s_i=s', s_{i+1}=s, y_0^{n-1})}{p(y_0^{n-1})}}{\sum_{(s',s)\in S_1} \frac{p(s_i=s', s_{i+1}=s, y_0^{n-1})}{p(y_0^{n-1})}}$$

где S_0 и S_1 – множества пар состояний $s' \in V_i, s \in V_{i+1}$, переход между которыми помечен соответственно 0 и 1, $p(y_0^{n-1})$ – 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 символа определяется только его состоянием s' на предыдущем шаге; канал не имеет памяти

$$p(s_i = s', s_{i+1} = s, y_0^{n-1}) = p(s_i = s', y_0^{i-1}) p(s_{i+1} = s, y_i | s_i = s', y_0^{i-1}) p(y_{i+1}^{n-1} | s_{i+1} = s, s_i = s', y_0^i) =$$
$$= \underbrace{p(s_i = s', y_0^{i-1})}_{\alpha_i(s')} \underbrace{p(s_{i+1} = s, y_i | s_i = s')}_{\gamma_{i+1}(s',s)} \underbrace{p(y_{i+1}^{n-1} | s_{i+1} = s)}_{\beta_{i+1}(s)}$$

Из формулы Байеса:

$$\alpha_i(s) = \sum_{\tilde{s} \in V_{i-1}} \alpha_{i-1}(\tilde{s}) \gamma_i(\tilde{s}, s), s \in V_i$$

$$\beta_i(\tilde{s}) = \sum_{s \in V_{i+1}} \gamma_{i+1}(s,s)\beta_{i+1}(s), s \in V_i$$

Непосредственное вычисление этих величин приводит к значительными ошибкам округления, поэтому приходится ввести вспомогательные величины $\alpha'_i(s) = \frac{\alpha_i(s)}{p(y_0^{i-1})}$ и $\beta'_i(s) = \frac{\beta_i(s)}{p(y_i^{n-1}|y_{i-1})}$. Поделив $p(s_i = s', s_{i+1} = s, y_0^{n-1})$ на $\frac{p(y_0^{n-1})}{p(y_i)} = p(y_0^i|y_i)p(y_{i+1}^{n-1}|y_0^i) = p(y_0^{i-1})p(y_{i+1}^{n-1}|y_0^i)$, получим

$$p(s_i = s', s_{i+1} = s|y_0^{n-1})p(y_i) = \frac{\alpha_i(s')\gamma_{i+1}(s',s)\beta_{i+1}(s)}{p(y_0^{i-1})p(y_{i+1}^{n-1}|y_0^i)} = \alpha'_i(s')\gamma_{i+1}(s',s)\beta'_{i+1}(s)$$

При этом

$$L_i = \ln \frac{\sum_{(s',s)\in S_1} \alpha'_i(s')\gamma_{i+1}(s',s)\beta'_{i+1}(s)}{\sum_{(s',s)\in S_0} \alpha'_i(s')\gamma_{i+1}(s',s)\beta'_{i+1}(s)}$$

Замечание. Хотим избавиться от ошибок округления

Учитывая, что $p(y_0^{i-1}) = \sum_{x\in V_i} \alpha'_i(s)$, получим

$$\alpha'_i(s) = \frac{\alpha'_i(s)}{\sum_{s'\in V_i} \alpha'_i(s')} = \frac{\sum_{\tilde{s}\in V_{i-1}} \alpha'_{i-1}(\tilde{s})\gamma_i(\tilde{s},s)}{\sum_{s'\in V_i}\sum_{\tilde{s}\in V_{i-1}} \alpha'_{i-1}(\tilde{s})\gamma_i(\tilde{s},s')}$$

Начальные условия: $\alpha'_0(s) = \alpha_0(s) = \begin{cases} 1 & , s = 0 \\ 0 & , s \neq 0 \end{cases}$

$$p(y_i^{n-1}|y_0^{i-1}) = p(y_i^{n-1}|y_0^{i-1})\frac{p(y_0^{i+1})}{p(y_0^{i-1})} = p(y_{i+1}^{n-1}|y_0^{i+1})\frac{p(y_0^{i+1})}{p(y_0^{i-1})} = \frac{p(y_{i+1}^{n-1}|y_0^{i+1})}{p(y_0^{i-1})}p(y_0^{i+1}) =$$
$$= \frac{p(y_{i+1}^{n-1}|y_0^{i+1})}{p(y_0^{i-1})} \sum_{x\in V_{i+1}} \alpha_{i+1}(s) = p(y_{i+1}^{n-1}|y_0^{i+1}) \sum_{s\in V_i}\sum_{s'\in V_{i+1}} \alpha'_i(s')\gamma_{i+1}(s',s)$$

Отсюда вытекает что

$$\beta'_i(\tilde{s}) = \frac{\beta_i(s)}{p(y_i^{n-1}|y_{i-1})} = \frac{\sum_{s\in V_{i+1}} \gamma_{i+1}(\tilde{s},s)\beta'_{i+1}(s)}{\sum_{s\in V_i}\sum_{s'\in V_{i+1}} \alpha'_i(s')\gamma_{i+1}(s',s)}$$

Начальными значениями для этой рекуррентной формулы являются

$$\beta' n(s) = \beta_n(s) = \begin{cases} 1 & , s = 0 \\ 0 & , s \neq 0 \end{cases}$$

$$\gamma_{i+1}(s',s) = p(s_{i+1} = s, y_i|s_i = s') = P\{s_{i+1} = s|s_i = s'\}p(y_i|s_i = s', s_{i+1} = s) =$$
$$= P\{c_i = \delta(s',s)\}p(y_i|c_i = \delta(s',s))$$

Вероятность $P\{c_i = \delta(s',s)\}$ представляет собой априорную вероятность того, что этот бит равен метке $\delta(s',s)$ перехода между состояниями s', s

Если рассматриваемый декодер используется как часть итеративного декодера составного кода, эта вероятность может быть найдена из апостериорных логарифмических отношений

правдоподобия $L_i^{(e)}$, вычисленных другим декодером, как $P\{c_i = 1\} = \frac{\exp(L_i^{(e)})}{1+\exp(L_i^{(e)})}$, откуда следует, что

$$P\{c_i = a\} = \frac{\exp\left(\frac{L_i^{(e)}}{2}\right)}{1+\exp(L_i^{(e)})} \exp\left((2a-1)\frac{L_i^{(e)}}{2}\right)$$

В противном случае все символы можно считать равновероятными, что эквивалентно $L_i^{(e)} = 0$

- Нахождение логарифмических отношений правдоподобия отдельных символов кодового слова $L(c_i), i = 0 \dots n-1$
- Вычисление величин $\gamma_k(s', s) = P\{c_i = \delta(s', s)\}p(y_i|c_i = \delta(s', s))$
- Вычисление $\alpha'_i(s)$ (прямая рекурсия) согласно $\alpha'_i(s) = \frac{\sum_{\tilde{s}\in V_{i-1}} \alpha_{i-1}(\tilde{s})\gamma_i(\tilde{s},s)}{\sum_{s'\in V_i}\sum_{\tilde{s}\in V_{i-1}} \alpha_{i-1}(\tilde{s})\gamma_i(\tilde{s},s')}$, $0 < i \leq n$
- Вычисление $\beta'_i(s)$ (обратная рекурсия) согласно $\beta'_i(\tilde{s}) = \frac{\sum_{s\in V_{i+1}} \gamma_{i+1}(\tilde{s},s)\beta'_{i+1}(s)}{\sum_{s\in V_i}\sum_{s'\in V_{i+1}} \alpha'_i(s')\gamma_{i+1}(s',s)}$, $0 \leq i < n$
- Вычисление апостериорных логарифмических отношений правдоподобия $L_i, 0 \leq i < n$, информационных битов согласно $L_i = \ln \frac{\sum_{(s',s)\in S_1} \alpha'_i(s')\gamma_{i+1}(s',s),\beta'_{i+1}(s)}{\sum_{(s',s)\in S_0} \alpha'_i(s')\gamma_{i+1}(s',s)\beta'_{i+1}(s)}$
- Принятие решения относительно каждого символа

Замечание. Полученная последовательность решений может не являться кодовым словом

Лекция 5

5.1 Сверточные коды

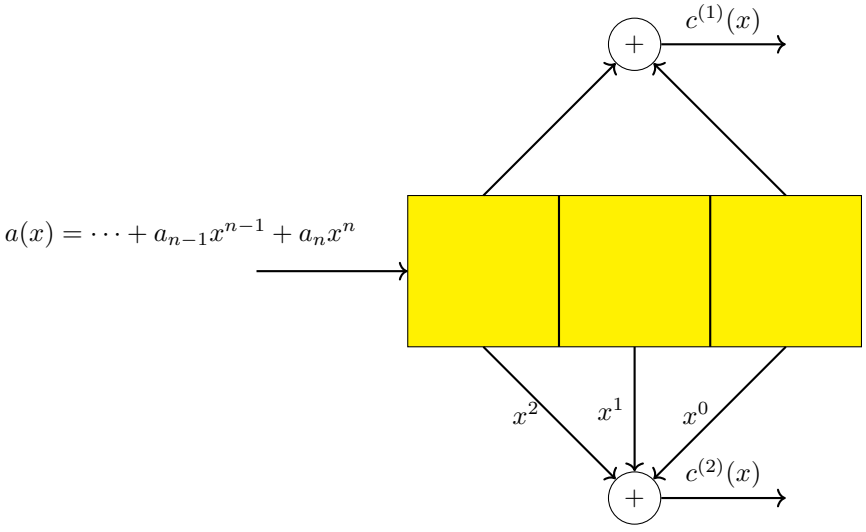
Замечание. Задача кодера – сделать передаваемые символы статистически зивисимыми

- Блоковые коды: функциональное преобразование блоков данных в кодовые слова
- Сверточные коды: автоматное отображение блоков данных в кадры колового слова

Простейший автомат – регистр сдвига. Кодер может хранить m ранее поступивших блоков из k_0 символов. На каждом шане кодер выдает n_0 закодированных символов. Скорость кода $R = \frac{k_0}{n_0}$. Объем памяти кодера – длина кодового ограничения $K = mk_0$.

Пример.

- $k_0 = 1, m = 2, K = k_0 m = 2, n_0 = 2$
- $g^{(1)}(x) = x^2 + 1, g^{(2)}(x) = x^2 + x + 1$



5.1.1 Порождающая матрица сверточного кода

Замечание. $k_0 = 1$: Выходная последовательность – линейная свертка информационной последовательности и порождащих многочленов кода

$$c^{(i)}(x) = c_0^{(i)} + c_1^{(i)}x + \cdots = a(x)g^{(i)}(x) = \sum_{j\geq 0} x^j \sum_{t=0}^{m} a_{j-t}g_t^{(i)}, 1 \leq i \leq n_0$$

Теоретически кодовые слова имеют бесконечную длину. В практических системах длина кодового слова фиксирована. В конец информационной последовательности вводят несколько дополнительных битов, переводящих регистр сдвига в нулевое состояние.

Кодирование в общем случае

$$(c^{(1)}(x), \ldots, c^{(n_0)}) = (a^{(1)}(x), \ldots, a^{(k_0)}(x))G(x)$$

$G(x)$ – $k_0 \times n_0$ порождающая матрица (многочленная) кода

Замечание. Сверточные коды являеются линейными

Замечание. Графическое представление сверточных кодов:

- Последовательности возможных переходов конечного автомата могут быть представлены в виде дерева. Древовидная диаграмма обладает свойством самоподобия
- Решетчатая диаграмма – боле компактный способ задания кода
- Кодовое слово – путь в решетке, начинающийся и заканчивающийся в нулевом состоянии. Фиксированная длина. Предполагается, что после обработки информационной последовательности были поданы несколько дополнительных битов, переводящих регистр сдвига в нулевое состояние

5.1.2 Корректирующая способность

Определение. Минимальное расстояние Хемминга для любых последовательностей из l кадров, отличающихся начальным кадром, называется **l-м минимальным расстоянием кода** d_l^*.

Обозначение. d_{m+1}^* – минимальное расстояние кода

Определение. Последовательность $d_1^*, d_2^*, d_3^*, \ldots$ называется **дистанционным профилем кода**

Утверждение. *Если в первых l кадрах произошло t ошибок, то первый кадр может быть исправлен при условии $2t + 1 \leq d_l^*$*

Определение. Минимальное свободное расстояние кода $d_{\text{frcc}} = \max_l d_l^*$

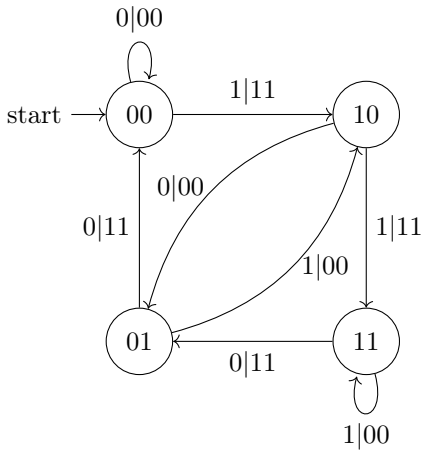
Замечание. Свободная длина n_{rfcc} кода – длина имеющего наименьший вес ненулевого начального сегмента кодовой последовательности сверточного кода

5.1.3 Катастрофические кодеры (порождающие матрицы)

Определение. Катастрофический кодер отображает информационные последовательности бесконечного веса в кодовые последовательности конечного веса

Замечание. КК характеризуется наличием петель нулевого веса в графе переходов

Пример. Порождающая матрица катастрофического кодера $G(x) = (x^2+1, x^2+1)$. Единичные ошибки в канале могут привести к бесконечному числу ошибок декодера. Если при передаче нулевого кодового слова возникла ошибка вида $\ldots 00011000 \ldots$, то она будет продекодирована в информационную последовательность вида $\ldots 000101010101 \ldots$, то она будет продекодирована в информационную последовательность вида $\ldots 000101010101 \ldots$



Замечание. Ошибка декодирования при исользовании некатастрофического кодера приводит к ограниченному числу ошибок на выходе декодера

Теорема 5.1.1. Порождающая матрица не является катастрофической ттогда, когда НОД определителей всех $k_0 \times k_0$ подматриц $G(x)$ равен $x^s, s \geq 0$.

5.1.4 Систематическое кодирование

Определение. Систематическое кодирование – информационная последовательность является подпоследовательностью кодовой последовательности

Любой сверточный код может быть преобразован к эквивалентному систематическому коду за счет введения фильтра с бесконечным импульсным откликом. Это преобразование информационной последовательности является биективным и не влияет на корректирующие свойства кода.

Утверждение. *Порождающая матрица $G(x)$ может быть приведена к каноническому виду аналогично случаю линейных блоковых кодов (Преобразование над полем рациональных функций)*

Некоторые информационные последовательности конечного веса могут попродить кодовые слова бесконечного веса

$$G(x) = (x^2 + 1, x^2 + x + 1) \to \left(\frac{x^2 + 1}{x^2 + x - 1}, 1\right)$$

Доделать Картинка

Пример. $a(x) = 1 + x^4 + x^5; a(x)G(x) = [1 + x + x^2 + x^5, 1 + x^4 + x^5]$

шаг	вход	содержимое регистра	выход 0	выход 1
0	1	100	1	1
1	1	010	0	1
2	0	101	0	0
3	0	110	1	0
4	0	011	1	0
5	1	001	1	1

5.2 Декодирование сверточных кодов

5.2.1 Алгоритм Витерби

Декодирование по критерию максимума праводоподобия (= минимального расстояния). Кодовые слова соответсвуют путям в решетке

Пример. Доделать Картинка

5.2.2 Производящая функция

Замечание. Вероятность ошибки декодирования кода определяется числом кодовых слов различного веса. Число путей в решетке, начинающихся и заканчивающихся в нулевом состоянии

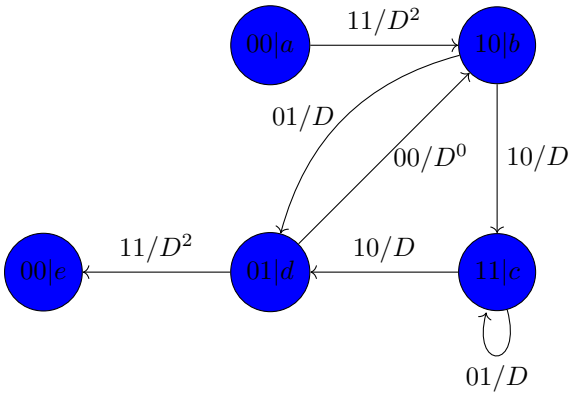
Пометим ребра графа переходов метками D^i, где i – вес кодовой последовательности. Последовательность символов веса x характеризуется одночленом D^x. Совокупность кодовых слов характеризуется многочленом, например $2D^6 + 3D^8$. Расщепим исходное состояние на два: начальное (0) и конечное (e). Пусть X_i характеризует совокупность кодовых последовательностей, приводящих кодер в состояние i. X_i – ряд, коэффиценты которого равны числу кодовых последовательностей, начинающихся в нулевом и заканчивающихся в i-ом состоянии. Производящая функция $T(D)$ равна X_e/X_a. Это степенной ряд, коэффиценты которого равны числу кодовых слов различного веса, выходящих из нулевого состояния и возвращающихся в него

Степень нунелувого члена – минимальное свободное расстояние кода. Минимальное свободное расстояние пропорционально длине кодового ограничения. При фиксированной длине блока сверточные коды хуже аналогичных блоковых

Пример.

$$\begin{cases} X_b = D^2 X_a + X_d \\ X_c = DX_b + DX_c \\ X_d = DX_c + DX_b \\ X_e = D^2 X_d \end{cases}$$

$$T(D) = \frac{D^5}{1 - 2D} = D^5 + 2D^6 + 4D^7 + 8D^8$$



5.2.3 Расширенная производящая функция

Исследуем зависимость веса кодового слова от веса информационной последовательности. Пометим ребра графа переходов метками $N^j D^i$, где i – вес выходной последовательности

- $j = 0$ – переход по 0
- $j = 1$ – переход по 1

Определение. Расширенная производящая функция – степенной ряд от переменных N и D, в котором коэффициент при $N^a D^b$ равен числу кодовых последовательностей веса b, порождаемых информационными последовательностями веса a.

Пример.

$$\begin{cases} X_b = ND^2X_a + ND^0X_d \\ X_c = NDX_b + NDX_c \\ X_d = DX_c + DX_b \\ X_e = D^2X_d \end{cases}$$

$$T(D) = \frac{ND^5}{1-2D} = ND^5 + 2N^2D^6 + 4N^3D^7 + 8N^4D^8$$

Доделать Картинка ez

5.2.4 Вероятность ошибки декодирования (канал с АБГШ)

Вероятность ошибки декодирования (канал с АБГШ)

$$r_{ij} = (-i)^{c_{ij}} + \eta_{ij}, \eta_{ij} \sim N(0, \sigma^2), 1 \le j \le n_0, i = 0, 1, \dots$$

Предположим, что передавалосб нулевое кодовое слово. Будем считать, что алгоритм Витерби ищет последовательность с максимальной корреляцией $C = \sum_{i \ge 0} \sum_{j=1}^{n_0} r_{ij}(-1)^{c_{ij}}$.

Оценим вероятность первого события неправильного декодирвания. Ошибка произойдет, если при слиянии нескольких путей на некотором ярусе B окажется, что $C_1 > C_0$. $_0$ – метрика ненулевого пути c_{ij}, C_1 – метрика нулевого пути

$$P\{C_1 > C_0\} = P\left\{\sum_{i=0}^{B}\sum_{j=1}^{n_0} r_{ij}((-1)^{c_{ij}} - 1) > 0)\right\} = P\left\{\sum_{i=0}^{B}\sum_{j:c_{ij}\neq 0} r_{ij} < 0\right\}$$

Объединенная верхняя граница вероятности ошибки декодирования:

- $r_{ij} \sum N(1, \sigma^2)$

Если неправильный путь имеет вес d на ярусах $0, \ldots, B$, то:

$$p = \sum_{i=0}^{B}\sum_{j:c_{ij}\neq 0} r_{ij} \sim N(d, d\sigma^2), \sigma^2 = \frac{N_0}{2}$$

$$P_d = P\{p < 0\} = Q\left(\sqrt{2d\frac{E_b}{N_0}}\right) = Q\left(\sqrt{2dR\frac{E_b}{N_0}}\right), Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-t^2/2}dt = \frac{1}{2}\text{erfc}(x/\sqrt{2})$$

Вероятность ошибки – вероятность того, что будет выбран какой-либо неправильный путь

$$P_C = P\{(C_1 > C_0) \vee (C_2 > C_0) \vee \ldots\} \leq \sum_i P\{C_i > C_0\} = \sum_{d>0} t_d P_d = \sum_{d=d?}^{\infty} t_d Q\left(\sqrt{2dR\frac{E_b}{N_0}}\right)$$

Производящая функция $T(D) = \sum_{d\geq 0} t_d D^d$

Вероятность ошибки на бит в случае выборп ошибочного пути C_i с кодовой и информационной последовательности отичающихся от истснных в d и w позициях, соответсвенно, равна $\frac{w}{k_0}P\{C_i > C_0\} = \frac{w}{k_0}P_d$. Расширенная производящая функция $T(N, D) = \sum_{w,d} t_{wd} N^w D^d$.

$$t(D) = \left.\frac{\partial T(N,d)}{\partial N}\right|_{N=1} = \sum_d D^d \underbrace{\sum_w t_{wd} w}_{b_d}$$

Общая вероятность ошибки декодирования на бит

$$P_b \leq \frac{1}{k_0}\sum_{d=d_{f?}}^{\infty} b_d Q\left(\sqrt{2dR\frac{E_b}{N_0}}\right)$$

5.2.5 Выводы

Сверточные коды – понятийно простой способ помехозащиты. Сложность декодирования алгритмом Витерби растет экспоненциально с длиной кодового ограничения и линейно с длиной кодируемой последовательности. Минимальное свободное расстояние растет с длиной кодового ограничения.

5.3 Комибинирование кодов

5.3.1 Конструкция Плоткина

Теорема 5.3.1. Пусть даны (n,k,d) коды $C_i, i=1,2$, $C=\{(c_1,c_1+c_2)|c_i\in C_i, i=1,2\}$ – $(2n, k_1+k_2, \min(2d_1,d_2))$ код

Доказательство. C содержит кодовые слова $(c_1,c_1), c_1\in C_1, (0,c_2), c_2\in C \implies d\le 2d_1, d\le d_2$

- Пусть $c_1,c_1'\in C_1\setminus\{0\}, c_2,c_2'\in C_2\setminus\{0\}$ – ненулевые кодовые слова компонентных кодов

$$d((c_1,c_1+c_2),(c_1',c_1'+c_2')) = d(c_1,c_1') + d(c_1+c_2,c_1'+c_2')$$

$$c_2=c_2'\wedge c_1\ne c_1' \implies d(c_1,c_1')+d(c_1+c_2,c_1'+c_2') = d(c_1,c_1')+d(c_1,c_1')\le 2d_1$$

$$\begin{aligned} c_2\ne c_2' \implies d(c_1,c_1')+d(c_1+c_2,c_1'+c_2') &= \mathrm{wt}(c_1-c_1')+\mathrm{wt}(c_1-c_1'+c_2-c_2')\ge \\ &\ge \mathrm{wt}(c_1-c_1')+\mathrm{wt}(c_2-c_2')-\mathrm{wt}(c_1-c_1') = \mathrm{wt}(c_2-c_2')\ge \\ &\ge d_2 \end{aligned}$$

□

5.3.2 Декодирование кодов

Декодирование $(y',y'')=(c_1,c_1+c_2)+(e',e'')$ в метрике Хемминга

$$y'''=y''-y'=c_1+c_2+e''-c_1-e'=c_2+e'''$$

Продекодируем y''' декодером кода C_2. Если $\mathrm{wt}((e',e''))\le\lfloor(d-1)/2\rfloor$, то $\mathrm{wt}(e''')\le\mathrm{wt}(e')+\mathrm{wt}(e'')\le\lfloor(d-1)/2\rfloor\le\lfloor(d_2-1)/2\rfloor$ и декодирование выполняется правильно&

Пусть c_2 найдено правильно. Продекодируем в C_1 вектора $y'=c_1+e'$ и $y''-c_2=c_1+e''$. Если $\mathrm{wt}((e',e''))=\mathrm{wt}(e')+\mathrm{wt}(e'')\le\lfloor(d-1)/2\rfloor\le\lfloor(2d_1-1)/2\rfloor<d_1$, то $\mathrm{wt}(e')\le\lfloor(d_1-1)/2\rfloor\vee\mathrm{wt}(e'')\le\lfloor(d_1-1)/2\rfloor \implies$ декодирование y' или y'' даст правильный результат

5.3.3 Коды Рида-Маллера

Замечание. Рекурсивное применение конструкции Плоткина

- $RM(r,m)$ – код Рида-Маллера порядка r длины 2^m
- $RM(0,m)=(2^m,1,2^m)$
- $RM(m,m)=(2^m,2^m,1)$
- $RM(r+1,m+1)$ применение конструкции Плоткина к $C_1=RM(r+1,m), C_2=RM(r,m)$

Лекция 6

6.1 Укорочение

Определение. Укороченный код полсучается путем выбора кодовых слов исходного кода, содержащих нули на заданных позициях, с последующим удалением этих нулей

Пусть дан (n, k, d) код с порождающей матрицей $G = (I|A)$. Удалим из порождающей матрицы m столбцев удиничной подматрицы и соответствующие m строк.

Пример. $(7, 4, 3)$ совершенный код, эквивалентный коду Хэмминга

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

$(5, 2, 3)$ совершенным не является

6.2 Выкалывание

Определение. Выколотый код: удалим из всех кодовых слов заданные символы (как правило, проверочных).

Пусть дана проверочная матрица (n, k, d) кода в форме $H = (A| - I)$. Удалим из H m столбцов единичной подматрицы и соответствующие им m строк. Если проявятся линейно зависимые строки, удалим их $\implies$ $(n - m, \leq k, \geq d - m)$ код

Пример. Построение оптимального кода $(10, 3, 5)$.

- Граница Грайсмера: $N(3, 5) \geq 5 + N(2, 3) \geq 5 + 3 + N(1, 2) = 5 + 3 + 2 = 10$
- C_1: Код Хемминга $(7, 4, 3)$ может быть укорочен до $(6, 3, 3)$
- C_2: Код с повторениями $(6, 1, 6)$
- Конструкция Плоткина $(C_1, C_1 + C_2)$: код $(12, 4, 6)$
- Выкалывание одного (последнего) символа: код $(11, 4, 5)$
- Укорочение на один символ: код $(10, 3, 5)$

$$G_{\text{Ham}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \Rightarrow \begin{matrix} G_1= \\ \\ G_2= \end{matrix} \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & & & & & & & & \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & & & & & & & & \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & & & & & & & & \end{pmatrix} \Rightarrow$$

$$\Rightarrow \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & & & & & & & & & \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & & & & & & & & & \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & & & & & & & & & \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

6.3 Расширение кодов

Определение. Наиболее распространенный способ – добавление проверки на четность. $(n, k, d) \Rightarrow (n + 1, k, d')$.

Если минимальное расстояние d исходного кода нечетно, то минимальное расстояние расширенного кода $d' = d + 1$.

Пример. $(7, 4, 3)$ код Хемминга

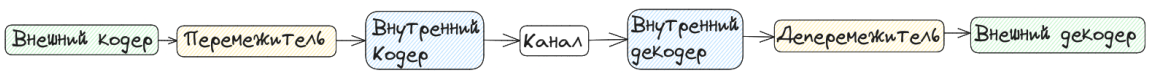
$$G = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}, H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$(8, 4, 4)$ расширенный код Хемминга

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

6.4 Каскадный код

Перемежитель таким образов переставляет символы, что последствия ошибочного декодирования одного кода могут быть легко ликвидированы декодером другого кода



6.5 Прямое произведение кодов

Определение. Пусть даны (n_1, k_1, d_1) (кодирование по строчкам) и (n_2, k_2, d_2) (кодирование по столбцам) коды с порождающими матрицами G', G''. Кодовое слово образуется путем выписывания полученной матрицы по столбцам. $(n_1 n_2, k_1 k_2, d_1 d_2)$ код с порождающей матрицей

$$G' \otimes G'' = \begin{pmatrix} G'_{11}G'' & G'_{12}G'' & \dots & G'_{1n_1}G'' \\ G'_{21}G'' & G'_{22}G'' & \dots & G'_{2n_1}G'' \\ \vdots & \vdots & \ddots & \vdots \\ G'_{k_1 1}G'' & G'_{k_1 2}G'' & \dots & G'_{k_1 n_1}G'' \end{pmatrix}$$

$$\mathrm{R} = \mathrm{k}_1 k_2 \overline{\frac{}{n_1 n_2 < \frac{k_1}{n_1} \cdot \frac{k_2}{n_2}}}$$

Код способен исправить многие конфигурации ошибок веса $> \frac{d_1 d_2}{2}$

Замечание. Параллельный алгоритм кодирования

Доделать Картинка

Пример. Код Рао-Редди $(48, 31, 8)$

Используется РЖД в рельсовых цепях сигнализации. Прямое произведение расширенного $(16, 11, 4)$ C_1 кода Хемминга и $(3, 2, 2)$ кода C_2 с проверкой на четность $\implies (48, 22, 8)$.

Замечание. Кодовые слова имеют вид $(c_1, c_2, c_1 + c_2), c_1, c_2 \in C_1$

Дополнение кодовыми словами (с дописанными в конец 32 нулями) кода Рида-Маллера $(16, 5, 8)$ C_3

Замечание. Кодовые слова имеют вид $(c_1 + c_3, c_2, c_1 + c_2), c_1, c_2 \in C_1, c_2 \in C_3$. Вес ненулевого кодового слова имеют вид $(c_1 + c_3, c_2, c_1 + c_2)$:

- $c_3 = 0 : \mathrm{wt}((c_1, c_2, c_1 + c_2)) \geq 8$
- $c_3 \neq 0 : \mathrm{wt}((c_1 + c_3, c_2, c_1 + c_2)) = \mathrm{wt}(c_1 + c_3) + \mathrm{wt}(c_2) + \mathrm{wt}(c_1 + c_2) \geq \mathrm{wt}((c_1 + c_3) + (c_2) + (c_1 + c_2)) = \mathrm{wt}(c_3) \geq 8$

Получен код $(48, 27, 7)$

Пусть C_1 – код с порождающей матрицей

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

Код $(48, 21, 8)$ Рао-Редди состоит из кодовых слова вида $(c_1+c_3+c_4, c_2+c_4, c_1+c_2+c_3), c_1, c_2 \in C_1, c_3 \in C_3, c_4 \in C_4$

6.6 TODO Лестничные коды

6.7 Каскадные коды

Закодируем данные внешним (N, K, D) кодом над $GF(q^k)$

Замечание. Строить хорошие коды над $GF(q^k), k > 1$ проще, чем над $GF(q)$

Пример. код Рида-Соломона с $D = N - K + 1$

Представим символы кодового слова как векторы длины k над $GF(q)$. Закодируем каждый символ (n, k, d) внутренним кодом над $GF(q)$. (Nn, Kk, Dd) код над $GF(q)$. Существуют каскадные коды, достигающие предела Шеннона для воичного симметричного канала

6.7.1 Граница Зяблова

Выберем внутренний (n, k, d) код на границе Варшамова-Гилберта с $r = \frac{k}{n} \geq 1 - h\left(\frac{d}{n}\right) = 1 - h(\delta)$. Внешний $(N, K, D = N - K + 1)$ код Рида-Соломона с $R = 1 - \frac{D-1}{N} \approx 1 - \Delta$. Сущесвтует код со скоростью $\rho = Rr$ и относительными расстоянием

$$\bar{\delta} = \frac{Dd}{Nn} = \Delta\delta \geq (1 - R)(1 - h^{-1}(r)) = \left(1 - \frac{\rho}{r}\right)(1 - h^{-1}(r))$$

$$\bar{\delta} \geq \max_{0 \leq r \leq h(\bar{\delta})} \left(1 - \frac{\rho}{r}\right)(1 - h^{-1}(r))$$

Далеко не все семейства кодов при длине $\to \infty$ одновременно обеспечивают скорость $\rho > 0$ и относительное минимальное расстояние $\bar{\delta} > 0$

6.7.2 Обобщенные каскадные коды

Определение. Внешние (N_i, K_i, D_i) коды $\mathcal{A}_i$ над $GF(q^{m_i}), 1 \leq i \leq s$.

Замечание. Вложенные внутренние (n_i, k_i, d_i) коды $\mathcal{B}_i : B_1 \supset \mathcal{B}_2 \supset \cdots \supset \mathcal{B}_s$ над $GF(q)$

- $k_i - k_{i+1} = m_i$
- Код $\mathcal{B}_i$ порождается последними k_i строками $k_1 \times n$ матрицы B

Кодирование:

- Закодируем данные внешними кодами и запишем полученные кодовые слова в $s \times N$ матрицу X
- Заменим элементы i-ой строки X на их векторное представление (столбец длиной m_i). Пусть Y – полученная $k_1 \times N$ матрица
- Умножим каждый столбец Y на B (т.е. закодируем в коде $\mathcal{B}_1$)
- Полученная $n \times N$ матрица может рассматриваться как кодовое слово

Линейный $(Nn, \sum_{i=1}^{s} K_i m_i, \geq \min_{1 \leq i \leq s} d_i D_i)$ код над $GF(q)$. Некоторые ОКК (полярные коды) достигают предела Шеннона.

Доделать Картинка

- Запишем принятые символы в виде $n \times N$ матрицы
- `for i=1, ... , s`
  - Продекодируем столбцы в коду $\mathcal{B}_i$
  - Продекодируем i-ую строку в коде $\mathcal{A}_i$. Пусть $(c_i, \ldots, c_N)$ – найденное кодовое слово
  - Вычтем из j-ого столбца $c_j B_i^T$

6.8 Турбо коды

Определение. Одновременное кодирование информационных битов нескольких сверточными кодами позволяет исключить многие конфигурации ошибок, которые могли бы возникнуть при использовании независимых сверточных кодов. Должны использоваться рекурсивные систематические сверточные коды. Многие информационные последовательности маого веса превращаются в кодовые слова большого веса. Длина кодового ограничения должна быть небольшой. Кодам с большим кодовым ограничением присущи длинные пакеты ошибок декодирования. Турбо-код является линейным блоковым кодом. Для повышения скорости кода используется выкалывание

Одновременное кодирование информационных битов нескольких сверточными кодами позволяет исключить многие конфигурации ошибок, которые могли бы возникнуть при использовании независимых сверточных кодов. Должны использоваться рекурсивные систематические сверточные коды. Многие информационные последовательности маого веса превращаются в кодовые слова большого веса. Длина кодового ограничения должна быть небольшой. Кодам с большим кодовым ограничением присущи длинные пакеты ошибок декодирования. Турбо-код является линейным блоковым кодом. Для повышения скорости кода используется выкалывание

Минимальное расстояние Доделать Картинка

Доделать Картинка

Декодеры сверточных кодов входящих в турбо-код, обмениваются информацией, полученной в результате декодирования. Как правило, достаточно $5 - 10$ итераций. Апостериорные логарифмические отношение правдоподобия информационных символов, вычисленные одним декодером, являются априорными ЛОПП для другого декодера. Аппроксимация декодера максимального правдоподобия. Этот подход применим и для декодирования прямого произведения кодов

6.8.1 Алгоритм Бала-Коке-Елинека-Равина декодирования светочных кодов

- Прямая рекурсия $\alpha_i'(s) = \frac{\sum_{\tilde{s} \in V_{i-1}} \alpha_{i-1}(\tilde{s}) \gamma_i(\tilde{s}, s)}{\sum_{s' \in V_i} \sum_{\tilde{s} \in V_{i-1}} \alpha_{i-1}(\tilde{s}) \gamma_i(\tilde{s}, s')}$, $0 < i \leq n$
- Обратная рекурсия $\beta_i'(\tilde{s}) = \frac{\sum_{s \in V_{i+1}} \gamma_{i+1}(\tilde{s}, s) \beta_{i+1}'(s)}{\sum_{s \in V_i} \sum_{s' \in V_{i+1}} \alpha_i'(s') \gamma_{i+1}(s', s)}$, $0 \leq i < n$
- Вычисление апостериорных ЛОПП $L_i = \ln \frac{\sum_{(s',s) \in S_1} \alpha_i'(s') \gamma_{i+1}(s', s), \beta_{i+1}'(s)}{\sum_{(s',s) \in S_0} \alpha_i'(s') \gamma_{i+1}(s', s) \beta_{i+1}'(s)}$
- Вероятности переходов (для кода со скоростью $1/2$)

$$\gamma_{i+1}(s',s) = p(s_{i+1}=s, y_{2i}, y_{2i+1}|s_i=s') = P\{s_{i+1}=s|s_i=s'\}p(y_{2i}, y_{2i+1}|s_i=s', s_{i+1}=s) = \\ = P\{u_i=\delta(s',s)\}p(y_{2i}, y_{2i+1}|(c_{2i}, c_{2i+1}) = \Delta(s',s))$$

- $P\{u_i=\delta(s',s)\}$ – априорная вероятность того, что информационный символ u_i принимает значение, соответствующее метке перехода $\delta(s',s) \in \{0,1\}$. В турбо-декодере эту вероятность вычисляет второй компонентный декодер
- $\Delta(s',s)$ – пара символов, формируемая кодером при переходе из состояния s' в s

6.8.2 Декодирование с использованием ЛОПП

- Внешние (extrinsic) ЛОПП $L_i^e = \ln\frac{P\{u_i=0\}}{P\{u_i=1\}} = \frac{P\{u_i=0\}}{1-P\{u_i=0\}}; P\{u_i=0\} = \frac{\exp(L_i^e)}{1+\exp(L_i^e)}$

$$P\{u_i=a\} = \begin{cases} \frac{\exp(L_i^e)}{1+\exp(L_i^e)}\exp(L_i^e/2) & ,a=0 \\ \frac{\exp(L_i^e)}{1+\exp(L_i^e)}\exp(-L_i^e/2) & ,a=1 \end{cases}$$

$$P\{u_i=a\} = A\exp((-1)^a L_i^e/2)$$

- Пусть $S_i = \ln\frac{P\{y_i|c_i=0\}}{P\{y_i|c_i=1\}}$. Вероятности выходных символов

$$p(y_{2i}, y_{2i+1}|(c_{2i}, c_{2i+1})) = B\exp\left(\frac{(-1)^{c_{2i}}S_{2i} + (-1)^{c_{2i+1}}S_{2i+1}}{2}\right)$$

- Коэффициенты A, B сокращаются во всех выражениях, используемых в алгоритме БКЕР, а потому могут быть отброшены

Получение внешних ЛОПП:
Предположим, что используется систематическое кодирование сверточных кодов и $c_{2i} = u_i$, где u_i – информационные символы. Результатом работы алгоритма БКЕР являются $L_i = \ln\frac{P\{u_i=0|y_0,\ldots,y_{2n-1}\}}{P\{u_i=1|y_0,\ldots,y_{2n-1}\}}$. Для турбо-декодирования необходимо вычислить $\tilde{L}_i^c = \ln\frac{P\{u_i=0|Y_{2i}\}}{P\{u_i=1|Y_{2i}\}}$, $Y_{2i} = (y_0,\ldots,y_{2i-1},y_{2i+1},\ldots,y_{2n-1}$. Это позволит исключить двойной учет принятых символов (по крайней мере, на первой итерации)

$$P\{u_i=a|Y_{2i},y_{2i}\} = \frac{P\{u_i=a,Y_{2i},y_{2i}\}}{P(Y_{2i},y_{2i})} = \frac{P(Y_{2i},y_{2i}|u_i=a)P\{u_i=a\}}{P(Y_{2i},y_{2i})} = \\ = \frac{P(Y_{2i}|u_i=a)P(y_{2i}|u_i=a)P\{u_i=a\}}{P(Y_{2i},y_{2i})} = \frac{P(Y_{2i}|u_i=a)P(y_{2i}|c_{2i}=a)P\{u_i=a\}}{P(Y_{2i},y_{2i})}$$

$$L_i = \ln\frac{P\{Y_{2i}|u_i=0\}}{P\{Y_{2i}|u_i=1\}} + S_i + L_i^e$$

Различные полуитерации используют различные Y_i, при этом изначально известно, что $P\{u_i=0\} = P\{u_i=1\} = 1/2$. Поэтому $\tilde{L}_i^c = L_i - S_i - L_i^e$
Итеративный алгоритм декодирования:

1. Положить $L_{1\to2}^e(u_i) = 0$

2. Воспользоваться декодером БКЕР для сверточного кодера 1
3. Подвергнуть перемежению полученные ЛОПП $\tilde{L}_i^e$ и использовать их как $L_{1\to 2}^e(u_i$
4. Воспользоваться декодером БКЕР для сверточного кодера 2
5. Подвергнуть деперемежению полученные апостериорные ЛОПП $\tilde{L}_i^e$ и использовать их как $L_{2\to 1}^e(u_i)$
6. Перейти к шагу 2, если не превышено максимальное число итераций $(5-10)$

6.8.3 Построение перемежителей

Требование: близкие позиции во входной последовательности должны отображаться в максимально удаленные позиции в выходной последовательности

$$0 < |i-j| < d \implies |\pi(i) - \pi(j)| \geq S$$

Важны объем памяти, требуемый для реализации перемежителя, его задержка. Псевдослучайный перемежитель: случайная генерация с отбрасыванием перестановок с неудовлетворительными S, d

Пример. Табличный перемежитель

$$(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16) \to \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} \to$$

$$\to (13,9,5,1,14,10,6,2,15,11,7,3,16,12,8,4)$$

Перестановочный полином: если $Q(x) = \sum_{i_0}^{l} q_i x^i \mod N$ – биекция в $\mathbb{Z}_N$, то перестановка может быть задана как $Q(i) \to Q(i+1$

Пример. $Q(x) = \frac{kx(x+1)}{2} \mod N, k \equiv 1 \mod 2$

6.9 Заключение

- Длинные коды можно строить из коротких
- Составные коды допускаюи простое декодирование
- Существуют каскадные коды
  - у которых относительное минимальное расстояние положительным при всех скоростях
  - достигающие предела Шеннона для двоичного симметричного канала (1966, Форни)
- Некоторые обобщенные каскадные коды (полярные) достигают предела Шеннона
- Турбо-коды стали первым классом корректирующих кодов, которые смогли на практике приблизится в пределу Шеннона (1993)

Лекция 7

7.1 Функция переходных вероятностей канала

Канал без памяти с входныи алфавитом

Определение. $W(y|c)$ – вероятность наблюдения на выходе канала $y \in \mathcal{Y}$ при условии подачи на его вход $c \in \mathcal{X}$

Пример. Двоичный симметричный канал: $\mathcal{Y} = \mathcal{X}, W(y|c) = \begin{cases} p & , y \neq x \\ 1-p & , y = x \end{cases}$

Пример. Двоичный стирающий канал $\mathcal{Y} = \{0, 1, \varepsilon\}$: $W(y|c) = \begin{cases} p & , y = \varepsilon \\ 1-p & , y = x \in \{0,1\} \end{cases}$

Пример. Двоичный симметричный канал со стираниями $\mathcal{Y} = \{0, 1, \varepsilon, W(y|c) = \begin{cases} 1-p-s & , y = x \\ s & , y = \varepsilon \\ p & , y = 1-x \end{cases}$

Непрервыный выходной алфавит $\mathcal{Y}$. $W(y|c)$ – плотность распределения выходного символа канала при подаче c на его вход. Аддитивный науссовский канал:
$\mathcal{Y} = \mathbb{R}, y = (-1)^c + \eta, \eta \sim \mathcal{N}(0, \sigma^2), W(y|c) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{|y-(-1)^c|^2}{2\sigma^2}}$ Для простоты, будем считать $\mathcal{Y}$ дискретным

7.2 Параметр Бхаттачарьи

Определение. Рассмотрим приемник по максимуму прадоподобия $\tilde{c} = \operatorname{argmax}_{c \in 0,1} W(y|c)$. Передаваемые символы рановероятны. Вероятность ошибки

$$\begin{aligned} P_c &= P\{c=0\}P\{err|e=0\} + P\{c=1\}P\{err|c=1\} = \\ &= \frac{1}{2} \sum_{y: W(y|0) < W(y|1)} W(y|0) + \frac{1}{2} \sum_{y: W(y|1) < W(y|0)} W(y|1) = \\ &= \frac{1}{2} \sum_{y: \frac{W(y|1)}{W(y|0)} > 1} W(y|0) + \frac{1}{2} \sum_{y: \frac{W(y|0)}{W(y|1)} > 1} W(y|1) = \\ &= \frac{1}{2} \sum_{y \in \mathcal{Y}} \sum_{c \in \{0,1\}} \left(W(y|c) \chi \left(\frac{W(y|1-c)}{W(y|c)} \right) \right) \end{aligned}$$

Индикаторная фунукция $\chi(z) = \begin{cases} 1 & , z \geq 1 \\ 0 & , z < 1 \end{cases}$

$$P_c \leq \frac{1}{2} \sum_{y \in \mathcal{Y}} \sum_{c \in \{0,1\}} \left(W(y|c) \chi \left(\frac{W(y|1-c)}{W(y|c)} \right) \right) = \sum_{y \in \mathcal{Y}} \sqrt{W(y|0)W(y|1)} = Z(W)$$

Пример. Двоичный стирающий канал

$$Z(BEC(p)) = \sqrt{W(0|0)W(0|1)} + \sqrt{W(1|0)W(1|1)} + \sqrt{W(\varepsilon|0)W(\varepsilon|1)} = p$$

Пример. Аддитивный гауссовский канал:

$$Z(\mathcal{G}(\sigma)) = \int_{-\infty}^{\infty} \sqrt{W(y|0)W(y|1)} dy = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\infty} e^{-\frac{(y-1)^2+(y+1)^2}{2\sigma^2}} dy = \frac{e^{-\frac{1}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\infty} e^{-\frac{y^2}{2\sigma^2}} dy = e^{-\frac{1}{2\sigma^2}}$$

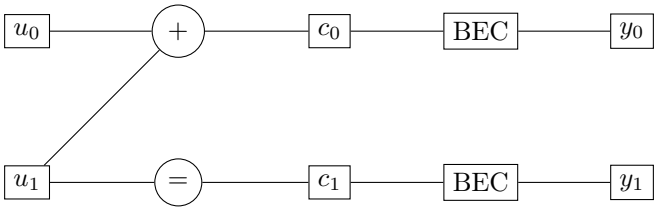
7.3 Пропускная способность канала

$$I(W) = \max_{\{p(x)\}} \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} W(y|x) P\{x\} \log \frac{W(y|x)}{W(y)}$$

Сущесутвуют методы сколь угодно надежной передачи данных со скоростью $R < I(W)$. При передаче данных со скоростью $R > I(W)$ вероятность ошибки ограничена снизу положительной величиной. Для многих каналов оптимальным распределением символов на входе $P\{x\}$ является равномерное

7.4 Поляризация канала

Определение. Рассмотрим линейное преобразование, задаваемое



$$\begin{pmatrix} c_0 & c_1 \end{pmatrix} = \begin{pmatrix} u_0 & u_1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

Двоичный стирающий канал: $y = \begin{cases} c_i & , \text{с вероятностью } 1-p \\ e & , \text{с вероятностью } p \end{cases}$

- u_0 не может быть восстановлен из y_0, y_1 с вероятностью $1 - (1-p)^2 = 2p - p^2 \geq p$
- u_1 не может быть восстановлен из u_0, y_0, y_1 с вероятностью $p^2 \leq p$

7.5 Битовые подканалы

Пусть $A_m = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{\otimes m}$, где $\otimes m$ обозначает m-кратное произведение Кронекерра матрицы с собой. Пусть $n = 2^m$.

Определение. Краткая запись подвекторов $y_a^b = (y_a, y_{a+1}, \dots, y_b)$

$$W_m(y_0^{n-1}|c_0^{n-1}) = \prod_{i_0}^{n-1} W(y_i|c_i)$$

Синтетические битовые подканалы

$$W_m^{(i)}(y_0^{n-1}, u_0^{i-1}|u_i) = \frac{W_m^{(i)}(y_0^{n-1}, u_0^i)}{P\{u_i\}} = 2 \sum_{u_{i+1}^{n-1} \in \mathbb{F}_2^{n-1}} W_m^{(n-1)}(y_0^{n-1}|u_0^{n-1}) P\{u_0^{n-1}\} =$$

$$= \frac{2}{2} \sum_{?} W_m(y_0^{n-1}|u_0^{n-1} A_m) = 2^{-n+1} \sum_{?} \prod_{?}^{n-1} W(y_j|(u_0^{n-1} A_m))$$ Исправить

7.6 Функция переходных вероятностей битовых подканалов

$$W_1^{(0)}(y_0, y_1|u_0) = \frac{1}{2} \sum_{u_1=0}^{1} W(y_0|u_0 + u_1) W(y_1|u_1)$$

$$W_1(y_0, y_1, u_0|u_1) = \frac{1}{2} W(y_0|u_0 + u_1) W(y_1|u_1)$$

7.7 Рекурсивное определение подканалов

$$W_\lambda^{2i}(y^{2^\lambda - 1}, u_0^{2i-1}|u_{2i}) = \frac{1}{2} \sum_{u_{2i+1}=0}^{1} W_{\lambda-1}^{(i)}(y_{0,\text{even}}^{2^\lambda-1}, u_{0,\text{even}}^{2i-1} + u_{0,\text{odd}}^{2i-1}|u_{2i} + u_{2i+1}) W_{\lambda-1}^{(i)}(y_{0,\text{odd}}^{2^\lambda-1}, u_{0,\text{odd}}^{2i-1}|u_{2i+1})$$

$$\mathrm{W}_\lambda^{(2i+1)}(y_0^{2^\lambda-1}, u_0^{2i}|u_{2i+1}) = \tfrac{1}{2} W_{\lambda-1}^{(i)}(y_{0,\text{even}}^{2^\lambda-1}, u_{0,\text{even}}^{2i-1} + u_{0,\text{odd}}^{2i-1}|u_{2i} + u_{2i+1}) W_{\lambda-1}^{(i)}(y_{0,\text{odd}}^{2^\lambda-1}, u_{0,\text{odd}}^{2i-1}|u_{2i+1})$$

Доделать Картинка ez

7.8 Параметры подканалов

Определение. Параметры Бхаттачарьи битовых подканалов $Z_{m,i} = Z(W_m^{(i)})$

$$Z_{m,2i+1} \leq Zm, 2i \leq 2Z_{m-1,i} - Z_{m-1,i}^2$$

$$Z_{m,2i+1} = Z_{m-1,i}^2$$

Строгое равенство в случае двоичного стирающего канала

Замечание. Пропускные способности битовых подканалов $I_{m,i} = I(W_m^{(i)})$

$$I_{m,2i} + I_{m,2i+1} = 2I_{m-1,i}$$

$$I_{m,2i} \leq I_{m,2i+1}$$

$$\sqrt{1 - Z(W)^2} \geq I(W) \geq \log \frac{2}{1 + Z(W)}$$

Для любого $\delta \in (0,1)$ при $m \to \infty$ доля подканалов с $I(W_m^{(i)}) \in (1-\delta, 1]$ стермиться к $I(W_0^{(0)}) - I(W)$, а доля подканалов с $I(W_m^{(i)}) \in [0, \delta)$ стремится к $1 - I(W)$

Замечание. Поляризация каналов: Доделать Картинка

- Доля недополяризованных подканалов убывает с увеличением m
- Число недополяризованных подканалов растет с увеличением m

7.9 Полярный код и алгоритм последовательного исключения

Замечание. Передавать предопределенные значения (например, 0) по плохим подканалам. Кодирование $c_0^{n-1} = u_0^{n-1} A_m, u_i = 0, i \in \mathcal{F}$, где $\mathcal{F}$ – множество номеров плохих подканалов (замороженных символов). Линейный блоковый код $(2^m, 2^m + |\mathcal{F}|)$

Program 2 Алгоритм последовательного исключения

1: **for** $i = 0, 1, \ldots, 2^m - 1$ **do**

2: $\hat{u}_i = \begin{cases} 0 & , i \in \mathcal{F} \\ \text{argmax}_{u_i} W_m^{(i)}(y_0^{n-1}, \hat{u}_0^{i-1} | u_i) & , i \notin \mathcal{F} \end{cases}$

3: **end for**

- Если предыдущие решения были правильными, то $\hat{u}_0^{i-1} = u_0^{i-1}$

Если ранее была допущена ошибка, алгоритм ПИ все равно не сможет ее исправить. Вероятность ошибки $P \leq \sum_{i \notin \mathcal{F}} Z_{m,i} \leq 2^{-n^\beta}, \beta < 0.5$

7.9.1 Сложность кодирования

$$u_0^{n-1} A_m = \begin{pmatrix} u_0^{n/2-1} & u_{n/2}^{n-1} \end{pmatrix} \begin{pmatrix} A_{m-1} & 0 \\ A_{m-1} & A_{m-1} \end{pmatrix} = \begin{pmatrix} (u_0^{n/2-1} + u_{n/2}^{n-1}) A_{m-1} & u_{n/2}^{n-1} A_{m-1} \end{pmatrix}$$

Сложность $T(n) = 2T(n/2) + n/2 = \frac{1}{2} n \log_2 n$

7.9.2 Декодер с ЛОПП

Логарифмическое отношение правдоподобия $L_m^{(i)}(y_0^{n-1}, u_0^{i-1}) = \ln \frac{W_m^{(i)}(y_0^{n-1}, u_0^{i-1}|0)}{W_m^{(i)}(y_0^{n-1}, u_0^{i-1}|1)}$

$$L_\lambda^{2i+1}(y_0^{n-1}, u_0^{i-1}) = \log \frac{W_{\lambda-1}^{(i)}(y_{0,\text{even}}^{2^\lambda-1}, u_{0,\text{even}}^{2i-1} + u_{0,\text{odd}}^{2i-1}|u_{2i}+0) W_{\lambda-1}^{(i)}(y_{0,\text{odd}}^{2^\lambda-1}, u_{0,\text{odd}}^{2i-1}|0)}{W_{\lambda-1}^{(i)}(y_{0,\text{even}}^{2^\lambda-1}, u_{0,\text{even}}^{2i-1} + u_{0,\text{odd}}^{2i-1}|u_{2i}+1) W_{\lambda-1}^{(i)}(y_{0,\text{odd}}^{2^\lambda-1}, u_{0,\text{odd}}^{2i-1}|1)} =$$
$$= (-1)^{u_{2i}} L_{\lambda-1}(y_{0,\text{even}}^{2^\lambda-1}, u_{0,\text{even}}^{2i-1} + u_{0,\text{odd}}^{2i-1}) + L_{\lambda-1}(y_{0,\text{odd}}^{2^\lambda-1}, u_{0,\text{odd}}^{2i-1})$$

Пусть

$$p_s = W_\lambda^{(2i)}(s|y_0^{2^\lambda-1}, u_0^{2i-1}) = \frac{W_\lambda^{(2i)}(y_0^{2^\lambda-1}, u_0^{2i-1}|s) P\{u_{2i}=s\}}{W_\lambda^{(2i)}(y_0^{2^\lambda-1}, u_0^{2i-1})} = \frac{W_\lambda^{(2i)}(y_0^{2^\lambda-1}, u_0^{2i-1}|s)}{2W_\lambda^{(2i)}(y_0^{2^\lambda-1}, u_0^{2i-1})}$$
$$p_{0s} = W_{\lambda-1}^{(i)}(s|y_{0,\text{even}}^{2^\lambda-1}, u_{0,\text{even}}^{2i-1} + u_{0,\text{odd}}^{2i-1})$$
$$p_{1s} = W_{\lambda-1}^{(i)}(s|y_{0,\text{even}}^{2^\lambda-1}, u_{0,\text{even}}^{2i-1}), s \in \{0,1\}$$
$$p_0 = p_{00}p_{10} + p_{01}p_{11}$$
$$p_1 = p_{01}p_{10} + p_{00}p_{11}$$

$$p_0 + p_1 = 1, p_{i0} + p_{i1} = 1, i \in \{0,1\}$$

$$\tanh\left(\frac{1}{2}\ln\frac{p_0}{p_1}\right) = \frac{\exp(\ln(p_0/p_1)) - 1}{\exp(\ln(p_0/p_1)) + 1} = p_0 - p_1 = 1 - 2p_1$$

$$1 - 2p_1 = (1-2p_{01})(1-2p_{11}) = 1 - 2(p_{01} + p_{11} - 2p_{11}p_{01}) = 1 - 1(p_0(1-p_{11}) + (1-p_{01})p_{11})$$

$$\tanh\left(\frac{1}{2}L_\lambda^{(2i)}(y_0^{n-1}, u_0^{2i-1})\right) = \tanh\left(\frac{1}{2}L_{\lambda-1}^{(i)}(y_{0,\text{even}}^{2^\lambda-1}, u_{0,\text{even}}^{2i-1} - u_{0,\text{odd}}^{2i-1})\right)\tanh\left(\frac{1}{2}L_{\lambda-1}^{(i)}(y_{0,\text{odd}}^{2^\lambda-1}, u_{0,\text{odd}}^{2i-1})\right)$$

$$L_\lambda^{(2i)}(y_0^{2^\lambda-1}, u_0^{2i-1}) = 2\tanh^{-1}\left(\tanh\left(\frac{1}{2}L_{\lambda-1}^{(i)}(y_{0,\text{even}}^{2^\lambda-1}, u_{0,\text{even}}^{2i-1} + u_{0,\text{odd}}^{2i-1})\right)\tanh\left(\frac{1}{2}L_{\lambda-1}^{(i)}(y_{0,\text{odd}}^{2^\lambda-1}, u_{0,\text{odd}}^{2i-1})\right)\right)$$

$$L_\lambda^{(2i+1)}(y_0^{n-1}, u_0^{2i}) = (-1)^{u_{2i}} L_{\lambda-1}(y_{0,\text{even}}^{2^\lambda-1}, u_{0,\text{even}}^{2i-1} + u_{0,\text{odd}}^{2i-1}) + L_{\lambda-1}(y_{0,\text{odd}}^{2^\lambda-1}, u_{0,\text{odd}}^{2i-1})$$

Program 3 Алгоритм последовательного исключения с ЛОПП

1: **for** $i = 0, 1, \ldots, 2^m$ **do**
2: $\hat{u}_i = \begin{cases} 0 & , i \in \mathcal{F} \\ 0 & , L_m^{(i)}(y_0^{n-1}, \hat{u}_0^{i-1}) > 0, i \notin \mathcal{F} \\ 1 & , L_m^{(i)}(y_0^{n-1}, \hat{u}_0^{i-1}) \le 0, i \notin \mathcal{F} \end{cases}$
3: **end for**

7.9.3 Другой вариант алгоритма последовательного исключения

$$W_m^{(i)}(u_0^i|y_0^{n-1}d$$

$$2\mathrm{W}(\mathrm{y}_0^{n-1}) = \omega \sum_{u_{i+1}^{n-1}} \prod_{j=0}^{n-1} W((u_0^{n-1}A_m)|y_j)$$

$$W_\lambda^{(2i)}(u_0^{2i}|y_0^{n-1}) = \omega \sum_{u_{2i+1}} W_{\lambda-1}^{(i)}(u_{2t} + u_{2t+1}, 0 \le t \le i|y_{0,\mathrm{even}^{n-1}})W_{\lambda-1}(u_{2t+1}, 0 \le t \le i|y_{0,\mathrm{odd}}^{n-1})$$

$$W_\lambda^{(2i+1)}(u_0^{2i-1}|y_0^{n-1}) = \omega W_{\lambda-1}^{(i)}(u_{2t} + u_{2t+1}, 0 \le t \le i|y_{0,\mathrm{even}^{n-1}})W_{\lambda-1}(u_{2t+1}, 0 \le t \le i|y_{0,\mathrm{odd}}^{n-1})$$

Доделать Картинка: зеленый – принятый вектор

Переиспользование на фазе $2i+1$ сомножителей $W_{\mu-1}^{(i)}$, вычисленных на фазе $2i$. При обновлени на слое λ вычисляют $2^{m-\lambda}$ ЛОПП. Сложность $C = \sum_{\lambda=1}^{m} 2^\lambda \cdot 2^{m-\lambda} = m2^m = n\log_2 n$. Сложность $O(n\log_2 n)$. Размер памяти $O(n)$.

7.9.4 Построение $(2^m, k)$ полярного кода

Замораживанию подлежат $2^m - k$ наименее надежных символов (например, с наибольшим $Z_{m,i}$. Двоичный стирающий канал

$$Z_{m,2i} = 2Z_{m-1,i} - Z_{m-1,i}^2$$

$$Z_{m,2i+1} = Z_{m-1,i}^2$$

Сложность вычисления $Z_{m,i} = O(n)$. В общем случае выходной алфавит канала $W_m^{(i)}(y_0^{n-1}, u_0^{i-1}|u_i)$ имеет мощность $|\mathcal{Y}|^n 2^i$. Построение функции переходных вероятностей $W_m^{(i)}$ вычислительно нереализуемо уже при небольших m. Можно аппроксимировать канал $W_m^{(i)}$ каналом с выходным алфавитом фиксированной мощности μ, который был бы чуть лучше или чуть хуже, чем истинный $W_m^{(i)}$. $Z_{m,i}$ могут быть вычисленны со сложностью $O(n\mu^2\log\mu)$.

7.9.5 Гауссовская аппроксимация

Полярные коды являются линейными. Для симметричных канало вероятность ошибки не зависит от того, какое кодовое слова передавалось. Будем считать, что передавалось 0 слово. Рассмотрим передачу кодовых слов по аддитивному гауссовскому каналу:
$y_i = (-1)^{c_i} + \eta_i, \eta_i \sim \mathcal{N}(0, \sigma^2) \implies L_0^{(0)}(y_i) = \frac{2y_i}{\sigma^2}$. Т.к. все $c_i = 0$.
$M[L_0^{(0)}(y_i)] = \mu_{00} = \frac{2}{\sigma^2}$. $D[L_0^{(0)}(y_i)] = \frac{4}{\sigma^2} = 2M[L_0^{(0)}(y_i)]$

Предположим, что все ЛОПП имеют нормальное распределние $\mathcal{L}_\lambda^{(i)} \sim N(\mu_{\lambda,i}, 2\mu_{\lambda,i}), 0 \le i < 2^\lambda, 0 \le \lambda \le m$

$$\mu_{\lambda,2i} = \Theta(\mu_{\lambda-1,i}) = \phi^{-1}\left(1 - (1 - \phi(\mu_{\lambda-1,i}))^2\right)$$

$$\mu_{\lambda,2i+1} = 2\mu_{\lambda-1,i}$$

$$\phi(x) = 1 - \frac{1}{\sqrt{4\pi x}} \int_{-\infty}^{\infty} \tanh\frac{u}{2} e^{-\frac{(u-x)^2}{4x}} du$$

Замораживаются символы с наименьшим $\mu_{m,i}$

Пример. Кусочно-квадратичная аппроксимация

$$\Theta(x) \approx \begin{cases} 0.9861x - 2.3152 & , x > 12 \\ x(9.005 \cdot 10^{-3}x + 0.7694) - 0.9507 & , x \in (3.5, 12] \\ x(0.062883x + 0.3678) - 0.1627 & , x \in (1, 3.5) \\ x(0.2202x + 0.066448) & , \text{иначе} \end{cases}$$

Доделать Картинка

7.10 Конструкция Плоткина и коды Рида-Маллера

Теорема 7.10.1. Пусть даны (n, k_i, d_i) коды $C_i, i = 0, 1$. $C = \{(c_1 + c_0, c_1) | c_i \in C_i\}$ – код $(2n, k_1 + k_0, \min(2d_1, d_0))$

Определение. Код Рида-Маллера $RM(r, m)$ длины 2^m порядка r – полярный код с $\mathcal{F} = \{i | 0 \leq i < 2^m, wt(i) < m - r\}$

Размерность $k = \sum_{i=m-r}^{m} C_m^i = \sum_{i=1}^{r} C_m^i$. Минимальное расстояние $d = 2^{m-r}$

Теорема 7.10.2. Минимальное расстояние d полярного кода длины $n = 2^m$ с замороженным множеством $\mathcal{F}$ равно $\min_{i \notin \mathcal{F}} 2^{wt(i)} = \min_{i \notin \mathcal{F}} wt(A_m^{(i)})$, где вес целого числа – число его ненулевых битов, $A_m^{(i)}$ – i-ая строка A_m

7.11 Минимальное расстояние кодов Рида-Маллера, БЧХ и полярных

Доделать Таблица

Замечание. Полярные коды не фонтан

Лекция 8

8.1 Субоптимальность полярных кодов

Алгоритм последовательно исключения

Program 4 Алгоритм последовательного исключения

1: **for** $i = 0, 1, \ldots, 2^m$ **do**

2: $$\hat{u}_i = \begin{cases} 0 & , i \in \mathcal{F} \\ 0 & , L_m^{(i)}(y_0^{n-1}, \hat{u}_0^{i-1} > 0), i \notin \mathcal{F} \\ 1 & , L_m^{(i)}(y_0^{n-1}, \hat{u}_0^{i-1}) \leq 0, i \notin \mathcal{F} \end{cases}$$

3: **end for**

При принятии решения относительно незамороженного символа не учитываются ограничения замораживания на символы $u_j, j \in \mathcal{F}, j > i$. Если принято неправильное решение $\hat{u}_i$, алгоритм последовательного исключения не способен далее его исправить. Минимальное расстояние слишком мало [**org:** Проблема: При оцениывнии каждого $\hat{u}_i$ не учитываем замораживание на будущие символы

]

8.2 Списочное кодирование

Не будет принимать окончательное решение относительно u_i на фазе i. На каждой фазе i будем рассматривать L векторов u_0^{i-1}, строить их возможные продолжения u_0^i и выбирать из них L наиболее вероятных. Вектора u_0^i задают пути в кодовом дереве Доделать Картинка

8.2.1 Приближенный алгоритм декодирования

$$W_\lambda^{(i)}(u_0^i | y_0^{2^\lambda - 1}) = \frac{W_\lambda^{(i)}(y_0^{2^\lambda-1}, u_0^{i-1} | u_i)}{2W(y_0^{2^\lambda-1})} = \sum_{u_{i+1}^{2^\lambda-1}} W_\lambda^{(2^\lambda-1)}(u_0^{2^\lambda-1} | y_0^{2^\lambda-1})$$

$$W_\lambda^{(2i)}(u_0^{2i} | y_0^{2^\lambda-1}) = \sum_{u_{2i+1} d 0}^{1} W_{\lambda-1}^{(i)}(u_{0,\text{even}}^{2i+1} + u_{0,\text{odd}}^{2i+1} | y_{0,\text{even}}^{2^\lambda-1}) W_{\lambda-1}^{(i)}(u_{0,\text{odd}}^{2i+1} | y_{0,\text{odd}}^{2^\lambda-1})$$

$$W_\lambda^{(2i+1)}(u_0^{2i+1} | y_0^{2^\lambda-1}) = W_{\lambda-1}^{(i)}(u_{0,\text{even}}^{2i+1} + u_{0,\text{odd}}^{2i+1} | y_{0,\text{even}}^{2^\lambda-1}) W_{\lambda-1}^{(i)}(u_{0,\text{odd}}^{2i+1} | y_{0,\text{odd}}^{2^\lambda-1})$$

Наиболее вероятное продолжение пути u_0^i (без учета замороженных символов $u_j = 0, j \in \mathcal{F}, j > i$)

$$\tilde{W}_\lambda^{(i)}(u_0^i|y_0^{2^\lambda-1}) = \max_{u_{i+1}^{2^\lambda-1}} W_\lambda^{(2^\lambda-1)}(u_0^{2^\lambda-1}|y_0^{2^\lambda-1})$$

$$\tilde{W}_\lambda^{(2i)}(u_0^{2i}|y_0^{2^\lambda-1}) = \max_{u_{2i+1}} \tilde{W}_{\lambda-1}^{(i)}(u_{0,\text{even}}^{2i+1} - u_{0,\text{odd}}^{2i+1}|y_{0,\text{even}}^{2^\lambda-1})\tilde{W}_{\lambda-1}^{(i)}(u_{0,\text{odd}}^{2i+1}|y_{0,\text{odd}}^{2^\lambda-1})$$

$$\tilde{W}_\lambda^{(2i+1)}(u_0^{2i+1}|y_0^{2^\lambda-1}) = \tilde{W}_{\lambda-1}^{(i)}(u_{0,\text{even}}^{2i+1} + u_{0,\text{odd}}^{2i+1}|y_{0,\text{even}}^{2^\lambda-1})\tilde{W}_{\lambda-1}^{(i)}(u_{0,\text{odd}}^{2i+1}|y_{0,\text{odd}}^{2^\lambda-1})$$

Определение. Вес пути в кодовом дереве

$$R_\lambda^{(i)}(u_0^i, y_0^{2^\lambda-1}) = \ln \tilde{W}_\lambda^{(i)}(u_0^i|y_0^{2^\lambda-1})$$

Модифицированное логарифмическое отношение правдоподобия

$$S_\lambda^{(i)}(u_0^{i-1}, y_0^{2^\lambda-1}) = R_\lambda^{(i)}(u_0^{i-1}.0, y_0^{2^\lambda-1}) - R_\lambda^{(i)}(u_0^{i-1}.1, y_0^{2^\lambda-1}) = \ln \frac{\tilde{W}_\lambda^{(i)}(u_0^{i-1}.0|y_0^{2^\lambda-1})}{\tilde{W}_\lambda^{(i)}(u_0^{i-1}.1|y_0^{2^\lambda-1})}$$

Если значение u_i соответствует наиболее вероятному продолжению пути u_0^{i-1}, то

$$R_m^{(i)}(u_0^i, y_0^{2^m-1}) = R_m^{(i-1)}(u_0^{i-1}, y_0^{2^m-1})$$

Если значение u_i не соответсвует наиболее вероятному продолжению пути u_0^{i-1}, то

$$R_m^{(i)}(u_0^i, y_0^{2^m-1}) = R_m^{(i-1)}(u_0^{i-1}, y_0^{2^m-1}) - (R_m^{(i)}(u_0^{i-1}.1 - u_i, y_0^{2^m-1}) - R_m^{(i)}(u_0^{(i-1)}.u_i, y_0^{2^m-1}))$$

$$R_m^{(i)}(u_0^i, y_0^{n-1}) = R_m^{(i-1)}(u_0^{i-1}, y_0^{n-1}) + \tau(S_m^{(i)}(u_0^{i-1}, y_0^{n-1}), u_i)$$

$$\tau(S, v) = \begin{cases} 0 & \text{, if } \operatorname{sgn}(S) = (-1)^v \\ -|S| & \text{, otherwise} \end{cases}$$

Модифицированные логарифмические отношения правдоподобия

$$\begin{aligned} S_\lambda^{(2i)}(u_0^{2i-1}|y_0^{2^\lambda-1}) &= max(J(0) + K(0), J(1) + K(1)) - max(J(1) + K(0), J(0) + K(1)) \\ &= max(J(0) - J(1) + K(0) - K(1), 0) - max(K(0) - K(1), J(0) - J(1)) \end{aligned}$$

где

$$J(c) = R_{\lambda-1}^{(i)}((v_{0,\text{even}}^{2i-1} \oplus v_{0,\text{odd}}^{2i-1}.c|y_{0,\text{even}}^{2^\lambda-1}), K(c) = R_{\lambda-1}^{(i)}(v_{0,\text{even}}^{2i-1}|y_{0,\text{even}}^{2^\lambda-1})$$

$$J(0) - J(1) = a = S_{\lambda-1}^{(i)}(v_{0,\text{even}}^{2i-1} \oplus v_{0,\text{odd}}^{2i-1}|y_{0,\text{even}}^{2^\lambda-1}), K(0) - K(1) = b = S_{\lambda-1}^{(i)}(v_{0,\text{odd}}^{2i-1}|y_{0,\text{odd}}^{2^\lambda-1})$$

8.2.2 Декодер min-sum

$$S_{\lambda}^{(2i)}(u_0^{2i-1}, y_0^{2^{\lambda}-1}) = Q(a,b) = \text{sgn}(a)\text{sgn}(b)\min(|a|,|b|)$$
$$S_{\lambda}^{(2i+1)}(u_0^{2i}, y_0^{2^{\lambda}-1}) = P(u_{2i}, a, b) = (-1)^{u_{2i}}a + b$$
$$a = S_{\lambda-1}^{(i)}(v_{0,\text{even}}^{2i-1} \oplus v_{0,\text{odd}}^{2i-1} | y_{0,\text{even}}^{2^{\lambda}-1}), b = S_{\lambda-1}^{(i)}(v_{0,\text{odd}}^{2i-1} | y_{0,\text{odd}}^{2^{\lambda}-1})$$

Program 5 Алгоритм последовательного исключения

1: **for** $i = 0, 1, \ldots, 2^m - 1$ **do**
2: $\hat{u}_i = \begin{cases} 0 & , i \in \mathcal{F} \\ 0 & , S_m^{(i)}(y_0^{n-1}, \hat{u}_0^{i-1} > 0), i \notin \mathcal{F} \\ 1 & , S_m^{(i)}(y_0^{n-1}, \hat{u}_0^{i-1}) \leq 0, i \notin \mathcal{F} \end{cases}$
3: **end for**

8.2.3 Списочный алгоритм Тала-Варди

- Пусть $V[l]$ – l-ый вектор u_0^{i-1}, рассматриваемый декодеромб $0 \leq l < L$
- Пусть $R[l] = R_m^{(i-1)}(V[l], y_0^{n-1})$

Program 6 Списочный алгоритм Тала-Варди

1: **for** $i = 0, 1, \ldots, 2^m - 1$ **do**
2: Вычислить $s_l = S_m^{(i)}(V[l], y_0^{n-1}), 0 \leq l < L$
3: **if** $i \in \mathcal{F}$ **then**
4: Дописать к $V[l]$ значение замороженного символа, $R[l] \leftarrow R[l] + \tau(s_l, (V[l])_i)$
5: **else**
6: Выбрать L пар (l, v) с наибольшим занчением r_{lv}, где $r_{lv} = R[l] + \tau(s_l, v), v \in \{0,1\}, 0 \leq l < L$
7: Для i-той выбранной пары (l, v) построить продолженный путь
8: $V'[i] \leftarrow V[i].v, R'[i] \leftarrow r_{lv}, 0 \leq i < L$
9: $V \leftarrow V', R \leftarrow R'$
10: **end if**
11: **end for**

Из полученного списка выбрать наилучший путь:

- С наибольшим $R[l]$
- Удовлетворяющий некоторым дополнительным условиям

8.2.4 Частичные суммы

Полярный код получается путем рекурсивного применения конструкции Плоткина $c = (c_1 + c_0, c_1)$. Массивы частичных сумм C_λ размерности $2^{m-\lambda}, 0 \leq \lambda \leq m$.

- В массиве C_0 размерности 2^m будем формировать кодовое слово c

- В массиве C_1 размерности 2^{m-1} будем офрмировать кодове слово c_0
- c_0 также является кодовым словом некоторого полярного кода. Соответствующий вектор $c_0 0$ будем формировать в C_2
- ...
- По готовности кодовых слов будем применять преобразование Плоткина и высвобождать массивы под кодовые слова следующий компонентных кодов

$u_0 = 0$

2	1	0
0		

$u_1 = 1$

2	1	0
0	$1 = 0 + 1$	
	1	

$u_2 = 1$

2	1	0
1	1	
	1	

$u_3 = 0$

2	1	0
1	1	$0 + 1 + 1$
	1	$0 + 1$
		$0 + 1$
		0

Массив ЛОПП S_λ размерности $2^{m-\lambda}, 0 \leq \lambda \leq m$. S_0 содержит ЛОПП принятого вектора. На i-той итерации алгоритма последовательного исключения обновляются массивы $S_{m-l}, \ldots, S_m$, где t – масимальная степень 2, делящая $i, t < m$

$$S_{m-t}[j] = P(C_{m-t-1}[j], Sm - y - 1[j], S_{m-t-1}[j + 2^t]), 0 \leq j < 2^{m-t}$$

$$S_\lambda[j] = Q(S_{\lambda-1}[j], S_{\lambda-1}[i + 2^{m-\lambda}]), m - t < \lambda \leq m$$

При $i = 0$ считается $t = m$, первая формула не используется

Замечание. Многие массивы частичных сумм и ЛОПП совпадают у различных путей. Копирование данных может быть исключено полностью

8.2.5 Полярные коды с CRC

Определение. Cyclic redundancy check – циклический код, обнаруживающий ошибки

Замечание. Систематическое кодирование циклического кода длины n с порождающим многочленом $g(x)$

$$c(x) = a(x)x^{n-k} + b(x), b(x) \equiv a(x)x^{n-k} \mod g(x)$$

Добавим к данным проверочные символы ($b(x)$ перед их кодированем полярным кодом. Удалим из списка, формируемого декодером Тала-Варди кодовые слова с направильным значением контрольной суммы.

8.2.6 Динамически замороженные символы

Классические полярные коды: замороженные символы $u_i = 0, i \in \mathcal{F}$. Обобщение: $u_i = \sum_{j=0}^{i-1} V_{s_i,j} u_j, i \in \mathcal{F}$

$$uV^T = 0$$

Program 7 Алгоритм последовательного исключения

1: **for** $i = 0, 1, \ldots, 2^m - 1$ **do**

2: $\hat{u}_i = \begin{cases} \sum_{j=0}^{i-1} V_{s_i,j} \hat{u}_j & , i \in \mathcal{F} \\ \text{argmax}_{u_i} W_m^{(i)}(y_0^{n-1}, \hat{u}_0^{i-1} | u_i & , i \notin \mathcal{F} \end{cases}$

3: **end for**

Вероятность ошибки (алгоритм ПИ) $P_{SC} \leq \sum_{i \notin \mathcal{F}} Z_{m,i}$. Совпадает с таковой для классических полярных кодов с тем же $\mathcal{F}$. Непосредственное обощение сисочного алгоритма Тала-Варди.

Как выбрать матрицу V:
Рассмотрим $(n = 2^m, k)$ линейный блоковый код с проверочной матрицей H. Пусть $V' = HA_m^T$. Применим к V' элементарные операции над строками, так, чтобы последние ненулевые элементы строк расположились в различных столбцах j_i. Пусть $V = QHA_m^T$ – полученная матрица. Алгоритм последовательного исключения можно использовать для декодирования произвольных линейных блоковых кодов.

8.2.7 Декодирование линейных кодов методов ПИ и его аналоги

- В общем случае вероятность ошибки декодирования методом ПИ намного больше вероятности ошибки декодирования по максимуму правдоподобия
- Перестановка столбцов проверочной матрицы (переход к эквивалентному коду) может радикально изменить множество $\mathcal{F}$ и P_{SC}
- Для расширенных примитивных кодов БЧХ в узком смысле P_{SC} достаточно мала $\Longrightarrow$ списочный алгоритм Тала-Варди с небольшим рамером списка может обеспечить хорошую корректирующую способность

1. Списочное декодирование полярных кодов и кодов БЧХ
   - Полярные коды минимизируют вероятность ошибки $P_{SC}(L = 1)$
   - При $L = 4$ обеспечивается декодирование полярных кодов по максимуму правдоподобия
   - При $L = 256$ обеспечивается декодирование кодо БЧХ почти по максимуму правдоподобия
   - Болшее минимальное расстояние обеспечивает лучшую корректирующую способность кодов БЧХ

8.2.8 Полярные коды в узком смысле

- Выберем линейный блоковый код (протокод) с достаточно большим минимальным расстоянием
- Удалим из него кодовые слова, препятствующие эффективному декодированию методом последовательного исключения

Определение. Рассмотрим канал $W(y|c)$ и $(n = 2^m, k', d)$ код $\mathcal{C}'$ над $GF(2)$, называемый протокодом. Пусть $\mathcal{F}'$ – множество номеров замороженных символов $\mathcal{C}'$. $(n, k, \geq d)$ полярным подкодом $\mathcal{C}$ в узком смысле кода $\mathcal{C}'$ называется множество векторов $c_0^{n-1} = u_0^{n-1} A_m$, где u_0^{n-1} одновременно удовлетворяет ограничениям замораживания кода $\mathcal{C}'$, а также дополнительным ограничениям $u_s = 0$ для $k' - k$ номеров $s \notin \mathcal{F}'$ с наибольшими вероятностями ошибки $P_{m,s}$.

Замечание. Расширенные примитивные коды БЧХ в узком смысле – хорошие протокоды

Матрица ограничений и матрица прекодирования:

$$c_0^{n-1} = u_0^{n-1} A_m, \quad u_0^{n-1} V^T = 0$$
$$u_0^{n-1} = xW, \quad WV^T = 0$$

- V – матрица ограничений (аналог проверочной матрицы)
- W – матрица прекодирования (аналог порождающей матрицы)

8.2.9 Полярные коды в широком смысле

- Выберем полярный код $(n = 2^m, n - r)$, эффективно декодируемый методом ПИ
- Удалим их него кодовые слова, ответственные за высокую вероятность ошибки декодирования МП, построив его подкод размерности $k < n - r$.

Определение. Полярным кодом в широком смысле называется множество векторо

$$c = x\mathbb{W}A_m, x \in FG(2)^k$$

где матрица $\mathbb{W}$ имеет нулевые стобцы в позициях, соответствующих r нименее надежным подканалам $W_m^{(j)}$

8.3 Выводы

- Полярные коды достигают предела Шеннон, имеют простые процедуры построения, кодирования и декодирования
- Корректирующая способность полярных кодов и алгоритма последовательного исключения неудовлетворительная
- Улучшенные декодеры: списочный и последовательный алгоритмы, метод распространения доверия
- Полряные подкоды и полярные коды с CRC на длинах до нескольких тысяч имеют лучшую корректирующую способность и меньшую сложность декодирования (при использовании последовательного декодирования) по сравнению с LDPC и турбо-кодами
- Алгоритм последовательного исключения и его аналоги плохо распараллеливаются и имеют большую задержку

Лекция 9

9.1 Группы

Определение. Группа $\mathcal{G}$ – алгебра $(G, \cdot)$.

- Операция $\cdot$ ассоциативна, т.е. $\forall a, b, c \in G : (a \cdot b) \cdot c = a \cdot (b \cdot c)$
- Существует нейтральный элемент $\mathbb{1} \in G : \forall x \in G : \mathbb{1} \cdot x = x \cdot \mathbb{1} = x$
- Существует обратный элемент $\forall x \in G \exists y : x \cdot y = y \cdot x = \mathbb{1}$

Замечание. Если $\cdot$ коммутативна, т.е. $a \cdot b = b \cdot a$, то группу называют коммутативной или абелевой. Далее все группы будем считать коммутативными

Определение. Аддитивное задание группы: $\cdot \to +, 1 \to 0$. Если вышеперечисленные свойства выполняются для некоторого $H \subset G$, замкнутого относительно операции $\cdot$, то $\mathcal{H} = (H, \cdot)$ называют подгруппой G.

Определение. Группы $\mathcal{G} = (G, \cdot)$ называется циклической, если $\exists a \in G : \forall x \in G \exists n \in \mathcal{Z} : x = a^n$

Определение.

- Порядок конечной группы – число элементов в ней, т.е. $|G|$
- Порядок элемента $a \in G$ – наименьшее положительное $n : a^n = \mathbb{1}$
- Порядок образующего элемента конечной циклической группы равен порядку самой группы

Теорема 9.1.1. Пусть $\mathcal{G} = (G, \cdot)$ – конечная группы и элементы $g, h \in G$ имеют порядок r, s соответственно, при чем $\gcd(r, s) = 1$. Тогда элемент gh имеет порядок rs.

Доказательство. То, что $(gh)^{rs} = \mathbb{1}$, очевидно. Следовательно, порядок p элемента gh – делитель числа rs. Пусть $p|(rs)$ и $(gh)^p = 1$. Тогда $(gh)^{pr} = h^{pr} = \mathbb{1}$. Следоваельно, $s|(pr)$, откуда $s|p$. Аналогично можно показать, что $r|p$. Т.к. $gcd(r, s) = 1$, получаем $(rs)|p$, что означает $p = rs$ $\square$

9.1.1 Подгруппы и смежные классы

Определение. Смежный класс по подгруппе $\mathcal{H}$: $g\mathcal{H} = \{g \cdot h | h \in G\}, g \in G$

Определение. Отношение эквивалентности $\sim_{\mathcal{H}} = \{(a, b) \in G^2 | a\mathcal{H} = b\mathcal{H}\}$. Класс эквивалентности $[a]_{\sim_{\mathcal{H}}} = a\mathcal{H}$

Лемма 1. *Всякий смежный класс подгруппы $\mathcal{H} = (H, \cdot)$ коммутативной группы $\mathcal{G} = (G, \cdot)$ равномощен H*

Доказательство. Для $a \in G$ зададим отображение $\phi_a : H \to a\mathcal{H}$ как $\phi_a(h) = ah$. Это сюръекция, т.к. $x \in \mathcal{H} \implies \exists h \in H : x = ah = \phi_a(h)$. Это инъекция, т.к. в группе $ah_1 = ah_1 \implies h_1 = h_2$. Следовательно, $\phi_a(h)$ – биекция и $|a\mathcal{H}| = |H|$ □

Теорема 9.1.2 (Лагранжа)**.** Порядок конечной группы $\mathcal{G} = (G, \cdot)$ делится на порядок любой ее подгруппы

Доказательство. Смежные классы $a\mathcal{H}$ подгруппы $\mathcal{H}$ группы $\mathcal{G}$ образуют разбиение G на равномощные подмножества □

9.2 Конечные поля

Определение. Полем называется алгебра $(\mathbb{F}, \{+, -, \cdot, /\})$, удовлетворяющая следующим аксиомам

- $a + (b + c) = (a + b) + c$
- $a + b = b + a$
- (∃ 0∈ F: a + 0= a\)
- $\forall a \in F : \exists a' = -a \in \mathbb{F} : a + a' = \mathbb{0}$
- $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
- $a \cdot b = b \cdot a$
- $a \cdot (b + c) + a \cdot b + a \cdot c, (b + c) \cdot a = b \cdot a + c \cdot a$
- $\exists \mathbb{1} \in \mathbb{F} \setminus \{\mathbb{0}\} : \forall a : a \cdot \mathbb{1} = \mathbb{1} \cdot a = a$
- $\forall a \neq \mathbb{0} \exists a^{-1} = \mathbb{1}/a \in \mathbb{F} : a \cdot a^{-1} = a^{-1} \cdot a = \mathbb{1}$

Замечание.

- Поля, содержащие конечное число q элементов – конечные поля или поля Галуа \(GF(q))
- Бесконечные поля называются полями характеристики нуль
- Поле является областью целостности, т.к. если допустить, что $\exists a, b \neq 0 : ab = \mathbb{0}$, то $\mathbb{0} = (((ab)b^{-1})a^{-1}) = \mathbb{1}) = 1$\)
- Простое поле $GF(p)$:
  - p – простое
  - арифметика по модулю p

9.2.1 Идеалы

Определение. Подмножество I кольца R называется **правым (или левым) идеалом**, если

- $(I, \{+\})$ является подгруппой $(R, \{+\})$
- $\forall r \in R, \forall x \in I : xr \in I$ (или $rx \in I$)

В коммутативных кольцах все идеалы являются двусторонними.

Определение. Если $A \subset R$, то идеалом, порождаемым A, называется нименьший идеал, содержащий A:

$$\langle A \rangle = \sum_i r_i a_i, a_i \in A, r_i \in R$$

Определение. Идеал I конечно порожден, если существует конечное множество $A : I = \langle A \rangle$

Определение. Идеал называется главным, если он порождается единственным элементом

Пример.

- Множество четных чисел является идеалом в кольце $\mathbb{Z}$, порожденным элементов 2
- Множество многочленов с вещественными коэффицентами, делащихся на x^2+1, является идеалом $\mathbb{R}[x]$
- Множество квадратных матриц, у которых последний столбец является нулевым, является левым идеалом в кольце квадратных матриц, но не является правым идеалом
- Кольцо непрерывных функций вещественного аргумента $C(\mathbb{R})$ содержит идеал непрерывных функций, таких что $f(1) = 0$
- $\{\mathbb{0}\}$ и R являются идеалами в любом кольце R
- $\langle x, y \rangle$ является идеалом, содержащим все многочлены от двух переменных с нулевым свободным членом

9.2.2 Максимальные идеалы

Область целостности, в которой все идеалы являются главными, называется кольцом главных идеалов

Определение. Идеал I кольца R называется максимальным, если $I \neq R$ и всякий идеал J, содержащий его, равен или I, или R

Идеал является максимальным, если он отличен от своего кольца и не содержится ни в каком ином идеале

Если $f_1, f_2, \ldots, f_n$ – система образующих масимального идеала I, то добавление в нее $f_0 \notin I$ приведет к $\langle f_0, \ldots, f_n \rangle = R$.

Пример. Идеал $\langle 7 \rangle \subset \mathbb{Z}$ является максимальным, т.к. числа, кратные 7, невозможно получить иначе как умножением 7 на целые числа

9.2.3 Факторкольца

Пусть дана полугруппа $\mathcal{G} = (G, \cdot)$. Бинарное отношение $\sim \subset G^2$ называется отношением конгруэнтности, если оно является эквивалентностью на G и $\forall a, b, c \in G : (a \sim b) \implies a \cdot c \sim b \cdot c$

Пример. $x \equiv y \mod n \Leftrightarrow x = qn + y$

Определение. Пусть дано кольцо R и идеал $I \subset R$. Бинарное отношение $\sim \equiv \{(a, b) \in R^2 | b - a \in I\}$ является отношением конгруэнтности. Разбиение на классы эквивалентности $[a] = a + I = a \mod I = \{a + r | r \in I\}$. Множество классов эквивалентности по отношению $\sim$ называется **факторкольцом** или кольцом вычетов R по модулю I и обозначается $R \setminus I$.

Замечание. Факторкольцо является кольцом, если на нем определить операции следующим образом:

- $(a + I) + (b + I) = \underbrace{(a + b)}_{\in R} + I$
- $-(a + I) = (-a) + I$
- $(a + I) \cdot (b + I) = ab + I$
- Нулевой элемент $\mathbb{0} + I = I$, единичный элемент $\mathbb{1} + I$

Если R – кольцо главных идеалов, и $\langle a \rangle \in R$, то соответствующее факторкольцо обозначают R/aR

Теорема 9.2.1. Если $I \subset R$ является максимальным идеалом, то R/I является полем

Доказательство. Надо показать, что для всякого ненулевого $a + I \in R/I$ существует $b + I : (a + I)(b + I) = \mathbb{1} + I$. Если $a + I \neq \mathbb{0}$, то $a \notin I$
Множество $J = \{ax + m | x \in R, m \in I\}$ является идеалом, т.к. $(ax_1 + m_1) \pm (ax_2 + m_2) = a(x_1 \pm x_2) + (m_1 \pm m_2), (x_1 \pm x_2) \in R, (m_1 \pm m_2) \in I$ и $(ax + m)y = a(xy) + (my), xy \in R, my \in I$ при любом $y \in R$.
Полагая $x = 0$ можно получить все элементы $I \implies I \subset J$.
Т.к. $a \in J, a \notin I$, а I – максимальный идеал, $J = R \implies \mathbb{1} \in J \implies \forall a \notin \mathbb{0} \exists b, m : \mathbb{1} = ab + m$, откуда $ab - 1 \in I$, т.е. $(a + I)(b + I) = \mathbb{1} + I$. Т.е. для каждого ненулевого элемента факторкольца по модулю максимального идеала существует обратный элемент, т.е. оно является полем □

Теорема 9.2.2. Пусть R – кольцо главных идеалов и p – его неприводимый элемент. Тогда факторкольцо R/pR является полем

Доказательство. Докажем, что $I = \langle p \rangle$ является масимальным. Предположим, что сущесвтует идеал $J \neq R : I \subset J$. Т.к. R – КГИ, J является главным и $\exists q \in R : J = \langle q \rangle$. Если q – обратимый элемент кольца, то $J = R$. Если $J \neq R$, то $p \in J \implies \exists s \in R : p = qs$. Но p – неприводимый элемент $\implies$ s – обратимый элемент кольца $\implies s^{-1} \in R \implies q = s^{-1}p \implies J \subset I = J$. Таким образом $\langle p \rangle = pR$ является масимальным и R/pR является полем. □

В дальшейшем подобные поля будут обозначаться просто R/p

Пример.
- Пусть p – простое число. $\mathbb{Z}$ является КГИ. Тогда $\mathbb{Z}/p$ является полем и обозначается $GF(p)$

- x^2+1 неприводим над $\mathbb{R}$. $\mathbb{R}[x]$ является КГИ. Следовательно $\mathbb{R}[x]/\left\langle x^2+1\right\rangle$ является полем (комплексным чисел $\mathbb{C}$).

$$(a+bx)(c+dx) = ac + (ab+bc)x + bdx^2 \equiv (ac-bd) + (ad+bc)x \mod (x^2+1)$$

  что соответсвует произведению комплексных чисел $a+ib$ и $c+id$, где $i = [x]_{\equiv_{x^2+1}}$. Видно, что $i \cdot i = x^2 \equiv -1 \mod (x^2+1)$

- x^2+x+1 неприводим над $GF(2)$. $GF(2)[x]$ является КГИ. Следовательно, $GF(2)/\left\langle x^2+x+1\right\rangle$ является полем и обозначается $GF(2^2)$ или $GF(4)$. Элементами $GF(4)$. Элементами $GF(4)$ являются классы вычетов многочленов $0, 1, x, x+1$ по модулю x^2+x+1. Все операции выполняются по модулю x^2+x+1. Например, $x \cdot (x+1) = x^2 + x \equiv 1 \mod (x^2+x+1)$. Аналогично $\mathbb{C}$, можно ввести $\beta \in GF(2^2)$, такой что $\beta^2 + \beta + 1 = 0$.

9.2.4 Характеристика поля

Определение. Если поле кончено, то не могут быть различными все элементы $\mathbb{1}, \mathbb{1}+\mathbb{1}, \mathbb{1}+\mathbb{1}+\mathbb{1}, \ldots$. Следовательно, существует наименьшее число $p : \underbrace{\mathbb{1}+\mathbb{1}+\cdots+\mathbb{1}}_{p \text{ раз}} = \mathbb{0}$. p – **характеристика поля**

p – простое число. Если это не так, т.е. $p = st$, то

$$\mathbb{0} = \underbrace{\mathbb{1}+\mathbb{1}+\cdots+\mathbb{1}}_{st \text{ раз}} = \underbrace{\underbrace{\mathbb{1}+\mathbb{1}+\cdots+\mathbb{1}}_{s \text{ раз}} + \cdots + \underbrace{\mathbb{1}+\mathbb{1}+\cdots+\mathbb{1}}_{s \text{ раз}}}_{t \text{ раз}} =$$

$$= (\underbrace{\mathbb{1}+\mathbb{1}+\cdots+\mathbb{1}}_{s \text{ раз}})(\underbrace{\mathbb{1}+\mathbb{1}+\cdots+\mathbb{1}}_{t \text{ раз}}) \implies s = p \vee t = p$$

Теорема 9.2.3. Пусть $\mathbb{F}$ – поле из q элементов. Тогда $q = p^m$, где p – простое, а $m \in \mathbb{N}$

Доказательство. Элемент $\mathbb{1} \in \mathbb{F}$ образует аддитивную циклическую подгруппу простого порядка p оля $\mathbb{F} \implies p|q \implies GF(p) \subset \mathbb{F}$. Будем называть элементы $\alpha_1, \ldots, \alpha_m$ линейно независимыми с коэффициентами из $GF(p)$, если $\{(x_1, x_2, \ldots, x_m) \in GF(p)^m | \sum_{i+1}^m x_i\alpha_i = 0\} = \{(0, \ldots, 0)\}$. Среди всех ЛНЗ подмножеств $\mathbb{F}$ выделим подмножество $\{\alpha_1, \ldots, \alpha_m\} \subset \mathbb{F}$ с максимальным числом элементов $\implies \forall \alpha_0 \in \mathbb{F} \exists x_1, \ldots, x_m \in GF(p) : \alpha_0 = \sum_{i=1}^m x_i\alpha_i$. Различные $x_1, \ldots, x_m \in GF(p)$ приводят к различным $\alpha_0 \in \mathbb{F} \implies |\mathbb{F}| = p^m$ □

Замечание.

- Поле $GF(q^m)$ образует m-мерное линейное пространство над полем $GF(q)$
- $GF(p^m), m > 1$ – расширенное конечное поле. m – степень расширения
- $GF(p^m), m > 1$ не имеет ничего общего с кольцом $\mathbb{Z}_{p^m}$ целых чисел по модулю p^m, которое полем не является

Теорема 9.2.4. Ненулевые элементы $GF(q)$ образует конечную циклическую группу по умножению

Доказательство.

- асиомы поля $\implies \mathbb{F} \setminus \{0\}$ образует конечную группу по умножению
- Выберем в $\mathbb{F} \setminus \{0\}$ элемента α (примитивный) с наибольшим порядком r. Пуст l – порядок некоторого другого элемента $\beta \neq 0$. Пусть π – простое число, такое что $r = \pi^a r'$ и $l = \pi^b l'$ и $\gcd(r', \pi) = \gcd(l', \pi) = 1$
- α^{π^a} имеет порядок r', а $\beta^{l'}$ имеет порядок π^b. Порядок $\gamma = \alpha^{\pi^a}\beta^{l'}$ равен $\pi^b r'$
- Т.к. r – наибольший порядок, $\pi^b r' \leq \pi^a r'$ и $b \leq a$. Это верно для всех простых $\pi \implies$ если некоторая степень простого числа является делителем l, то она является и делителем r. Следовательно, $l|r \implies$ все ненулевые элементы конечного поля удовлетворяет соотношению $x^r = 1$
- $x^r - \mathbb{1}$ принадлежит Евклидову кольцу $GF(q)[x]$ и разлагается на множители единственным образом $\implies \forall \beta \in GF(q) \setminus \{0\} : (x - \beta)|(x^n - 1)$, откуда $r \geq q - 1$
- Порядок элемента не может превышать порядка самой группы $\implies r = q - 1$

□

9.2.5 Свойства конечных полей

Замечание.

- Для всякого ненулевого $\beta \in GF(q)$ выполняется $\beta^{q-1} = 1$
- Все элементы поля $GF(q)$ удовлетворяют уравнению $x^q - x = 0$
- Порядок любого ненулевого $\beta \in GF(q)$ делит $q - 1$
- В поле характеристики $p > 1$ справедливо

$$(x + y)^p = x^p + y^p$$

$$(x + y)^p = \sum_{i=0}^{p} C_p^i x^i y^{p-1}; C_p^i = \frac{p!}{i!(p-i)!} \equiv 0 \mod p, 0 < i < p$$

Лекция 10

10.1 Минимальные многочлены

Определение. Минимальными многочленом элемента $\beta \in GF(p^m)$ над $GF(p)$ называется нормированный многочлен $M_\beta(x) \in GF(p)[x]$ наименьшей сетепени, т.ч. $M_\beta(\beta) = 0$

Теорема 10.1.1. $M_\beta(x)$ неприводим над $GF(p)$

Доказательство. Если $M_\beta(x) = M_1(x)M_2(x), \deg M_i(x) < \deg M(x), M_i(x) \in GF(p)[x]$ и $M_\beta(\beta) = 0$, то $M_1(\beta) = 0$ или $M_2(\beta) = 0 \implies$ степень $M_\beta(x)$ не минимальна □

Теорема 10.1.2. Если $f(x) \in GF(p)[x]$ и $f(\beta) = 0$, то $M_\beta(x)|f(x)$ ($f(x)$ делится на этот минимальный многочлен)

Доказательство.

$$f(x) = q(x)M_\beta(x) + r(x), 0 = f(\beta) = q(\beta)0 + r(\beta)$$

Теорема 10.1.3. $M_\beta(x)|(x^{p^m} - x)$ для $\beta \in GF(p^m)$

Доказательство. Утверждение непосредственно вытекает из предыдущей теоремы □

Теорема 10.1.4.

Доказательство. $GF(p^m)$ образует m-мерное линейное пространство над $GF(p) \implies$ любые $m+1$ элементов $GF(p^m)$ линейно зависимы над $GF(p)$. В частности $\forall\beta : \exists a_0, a_1, \ldots, a_m \in GF(p) : \sum_{i=0}^{m} a_i\beta^i = 0 \implies M(x) = \sum_{i=0}^{m} a_i x^i \in GF(p)[x]$ имеет корень β. Возможно, $M(x)$ можно разложить на сомножители меньшей степени. $M_\beta(x)|M(x)$, т.е. $\deg M_\beta(x) \le \deg M(x) \le m$ □

Теорема 10.1.5. Если α – примитивный элемент $GF(p^m)$, то степень его минимального многочлена равна m

Доказательство.

- Пусть $M_\alpha(x) = \pi(x) = \sum_{i=0}^{d} \pi_i x^i, \pi_i \in GF(p)$, причем $\alpha^d = -\sum_{i=0}^{d-1} \pi_i\alpha^i, d \le m$
- $\alpha^{d+1} = -\sum_{i=0}^{d-1} \pi_i\alpha^{i+1} = -\sum_{i=0}^{d-2} \pi_i\alpha^{i+1} + \pi_{d-1}\sum_{i=0}^{d-1} \pi_i\alpha^i = \sum_{i=0}^{d-1} a_{d+1,i}\alpha^i$
- Всякий $\beta \in GF(p^m) \setminus \{0\}$ может быть представлен как $\beta = \alpha^j = \sum_{i=0}^{d-1} a_{j,i}\alpha^i, a_{j,i} \in GF(p)$
- $GF(p^m)$ – m-мерное линейное пространство над $GF(p) \implies d \ge m$

□

Минимальные многочлен примитивного элемента поля называется примитивным. Не все неприводимые многочлены являются примитивными. Элементы $\beta \in GF(p^m)$ представимы как $\beta = \sum_{i=0}^{m-1} b_{\beta,i}\alpha^i, b_{\beta,i} \in GF(p)$

Теорема 10.1.6. Все конечные поля $GF(p^m)$ изоморфны

Доказательство.
- Пусть F и G – поля, содержащие p^m элементов
- Пусть α – примитивный элемент поля F с минимальным многочленом $\pi(x)$
- $\pi(x)|(x^{p^m} - x) \implies \exists\beta \in G : \pi(\beta) = 0$. Теперь F можно рассматривать как множество многочленов от α степени не более $m-1$, а G – как множество многочленов от β степени не боее $m-1$. Тогда соответсвие $\alpha \leftrightarrow \beta$ задает изоморфизм полей F и G

□

Пример. Рассмотрим два способа задания поля $GF(2^3)$

Через многочлен x^3+x+1	Через многочлен x^3+x^2+1
$(000) = 0$	$(000) = 0$
$(001) = 1 = \alpha^0$	$(001) = 1 = \gamma^0$
$(010) = \alpha$	$(010) = \gamma$
$(100) = \alpha^2$	$(100) = \gamma^2$
$(011) = \alpha^3 = \alpha + 1$	$(101) = \gamma^3 = \gamma^2 + 1$
$(110) = \alpha^4 = \alpha^2 + \alpha$	$(111) = \gamma^4 = \gamma^2 + \gamma + 1$
$(111) = \alpha^5 = \alpha^2 + \alpha + 1$	$(011) = \gamma^5 = \gamma + 1$
$(101) = \alpha^6 = \alpha^2 + 1$	$(110) = \gamma^6 = \gamma^2 + \gamma$

$\alpha^3 + \alpha + 1 = 0$. $(\gamma^3)^3 + \gamma^3 + 1 = \gamma^2 + \gamma^3 + 1 = 0$, т.е. $\pi(\alpha) = \pi(\gamma^3) = 0$, где $\pi(x) = x^3 + x + 1$. Таким образом, соответствие $\alpha \leftrightarrow \gamma^3$ задает изоморфизм между этими двумя полями

Теорема 10.1.7. $\forall\beta \in GF(p^m) : M_\beta(x) = M_{\beta^p}(x)$

Доказательство.

$$\mathbb{0} = M_\beta(\beta) = \sum_{i=0}^{d} M_{\beta,i}\beta^i, M_{\beta,i} \in GF(p)$$

$$\mathbb{0} = (M_\beta(\beta))^p = \sum_{i=0}^{d} M_{\beta,i}^p\beta^{pi} = \sum_{i=0}^{d} M_{\beta,i}\beta^{pi} = M_(\beta^p) \implies M_{\beta^p}(x)|M_\beta(x)$$

Т.к. минимальные многочлены неприводимы, $M_{\beta^p}(x) = M_\beta(x)$. $\beta, \beta^p, \ldots, \beta^{p^{m_\beta-1}}$ – сопряженные многочлены □

Теорема 10.1.8. $M_\beta(x) = \prod_{i=0}^{m_\beta-1}(x - \beta^{p^i})$, где m_β – наименьшее положительное число, т.ч. $\beta^{p^{m_\beta-1}} = \beta$

Доказательство. $M_(\beta) = 0$ – очевидно. $\prod_{i=0}^{m_\beta-1}(x - \beta^{p^i}) = \sum_{i=0}^{m_\beta} a_i x^i$.

$$\sum_{i=0}^{m_\beta} a_i^p x^{pi} = \left(\prod_{i=0}^{m_\beta-1}(x - \beta^{p^i})\right)^p = \prod_{i=0}^{m_\beta-1}(x^p - \beta^{p^{i+1}}) = \prod_{i+0}^{m_\beta-1}(x^p - \beta^{p^i}) = \sum_{i=0}^{m_\beta} a_i x^{pi} \implies a_i \in GF(p)$$

$M_\beta(x)$ имеет корни $\beta, \beta^p, \ldots, \beta^{p^{m_\beta - 1}} \implies$ предлагаемый многочлен имеет наименьшую возможную степень $\square$

10.2 Циклические коды

Определение. Линейный блоковый код $\mathcal{C}$ длины n над полем $\mathbb{F}$ называется циклическим, если любой циклический сдвиг его кодового слова также является кодовым словом, т.е. $(c_0, c_1, \ldots, c_{n-1}) \in \mathcal{C} \implies (c_{n-1}, c_0, c_1, \ldots, c_{n-1}) \in \mathcal{C}$

Замечание. Многочленное представление вектора $(c_0, c_1, \ldots, c_{n-1}) : c(x) = \sum_{i=0}^{n-1} c_i x^i$. Циклический свдиг вектора на одну позицию эквивалентен

$$xc(x) = xc_0 + x^2 c_1 + \cdots + c_{n-2}x^{n-1} + c_{n-1}x^n \equiv c_{n-1} + xc_0 + x^2 c_1 + \cdots + c_{n-2}x^{n-1} \mod x^n - 1$$

В дальнейшем вектор $(c_0, c_1, \ldots, c_{n-1})$ и соответствующей многочлен $c(x)$ будут считаться равнозначными

Теорема 10.2.1. Подмножество $\mathcal{C} \subset \mathbb{F}[x] \setminus (x^n - 1)$ образует циклический код ттогда, когда:

1. $\mathcal{C}$ образует группу по сложению
2. Если $c(x) \in \mathcal{C}$ и $a(x) \in \mathbb{F}[x] \setminus (x^n - 1)$, то $[a(x)c(x) \mod (x^n - 1)] \in \mathcal{C}$

Доказательство.

- Пусть $\mathcal{C}$ обладает указанными свойствами $\implies$
  - $\mathcal{C}$ замкнуто относительно операции умножения на скаляр $\implies$ образует линейное пространство
  - Умножение на x^i не выводит за пределы $\mathcal{C} \implies$ циклический код
- Пусть $\mathcal{C}$ – циклический код
  - линейный код по определению образует группу по сложению
  - Умножение на произвольный многочлен можно представить как взвешенную сумму циклических сдвигов

$\square$

10.2.1 Порождающий и проверочный многочлены

Замечание.

- Порождающий многочлен циклическоо кода – ненулевой кодовый многочлен $g(x) \in \mathcal{C}$ нименьшей степени с коэффициентов при старшем члене 1
- Все кодовые слова $c(x)$ в ЦК делятся на $g(x)$

  Предположим противное $\implies c(x) = a(x)g(x) + r(x), r(x) \in \mathcal{C}$. $\deg r(x) < \deg g(x)$, что противоречит предположению о минимальности степени $g(x)$
- Порождающий многочлен циклического кода единственен

- ЦК длины n с ПМ $g(x)$ сущесвует ттогда, когда $g(x)|(x^n-1)$
  - Существует код $\mathcal{C}$ с ПМ $g(x) \implies$
    - $x^n - 1 = a(x)g(x) + r(x), \deg r(x) < \deg g(x)$
    - $b(x) \equiv a(x)g(x) \mod (x^n-1), b(x) \in \mathcal{C}$
    - $r(x) = (x^n - 1 - a(x)g(x)) \equiv -a(x)g(x) \mod (x^n-1), r(x) \in \mathcal{C} \implies r(x) = 0$
  - $\Leftarrow$: в качестве порождающего многочлена можно выбрать любой делитель $x^n - 1$
- $(x^n-1) = h(x)g(x), h(x)$ – проверочный многочлен кода
- Для любого $c(x) \in \mathcal{C} : c(x)h(x) = a(x)g(x)h(x) \equiv 0 \mod x^n - 1$
- Размерность циклического кода равна $kd \deg h(x)$

10.2.2 Кодирование циклических кодов

Замечание. Несистематическое кодирование $c(x) = a(x)g(x)$

$$(c_0, \dots, c_{n-1}) = (a_0, \dots, a_{k-1}) \begin{pmatrix} g_0 & g_1 & g_2 & \dots & g_{n-k} & 0 & \dots & 0 \\ 0 & g_0 & g_1 & g_2 & \dots & g_{n-k} & \dots & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & \dots & g_{n-k} \end{pmatrix}$$

Замечание. Систематическое кодирование (информационные символы $a_0, \dots, a_{k-1}$ в $c_{n-k}, \dots, c_{n-1}$)

$$c(x) = x^{n-k}a(x) - r(x)$$

$$r(x) \equiv x^{n-k}a(x) \mod g(x), \deg r(x) < \deg g(x)$$

Каждому методу кодирования соответствует своя порождающая матрица. Все порождающие матрицы выражаются друг через друга как $G' = QG$, где Q – обратимая матрица. Используемый метод кодирования не влияет на корректирующую способность кода

10.2.3 Свойства порождающего многочлена

- $x^n - 1 = \prod_{i=0}^{l-1} f_i(x)$, где $f_i(x)$ – неприводимые над $GF(q)$ многочлены. Разложение однозначно с точностью до порядка записи сомножителей и их домножения на обратимые элементы
- $g(x)|(x^n-1) \implies g(x) = \prod_{i \in J} f_i(x), J \subset \{0, \dots, l-1\}$. Если все $f_i(x)$ различны , есть $2^l - 2$ нетривиальных циклических кода
- Циклические коды над $GF(q)$ длины $n = q^m - 1$ называются примитивными

Теорема 10.2.2. Пусть $\beta_1, \beta_2, \dots, \beta_r \in GF(q^m)$ – корни порождающего многочлена $g(x)$ примитивного циклического кода $\mathcal{C}$ над полем $GF(q)$. Многочлен $c(x) \in GF(q)[x]$ является кодовым тогда и только тогда, когда $c(\beta_1) = c(\beta_2) = \dots = c(\beta_r) = 0$

Доказательство. $c(x) = a(x)g(x) \implies$ все корни $g(x)$ являются корнями $c(x)$. $c(\beta_1) = 0 \implies M_i(x)|c(x)$, где $M_i(x)$ – минимальный многочлен β_i. Если $M_i(x)|c(x), i = 1, \dots, r \implies g(x)|c(x) \implies c(x) \in \mathcal{C}$ □

10.2.4 Проверочная матрица над расширенным полем

Замечание. Пусть порождающий многочлен циклического кода $\mathcal{C}$ над $GF(q)$ имеет корни $\beta_1, \ldots, \beta_r \in GF(q^m) \implies \forall c(x) \in \mathcal{C} : c(\beta_i) = 0, 1 \le i \le r \implies \sum_{j=0}^{n-1} c_j \beta_i^j = 0 \implies Gc^T = 0$

$$\begin{pmatrix} \beta_1^0 & \beta_1^1 & \ldots & \beta_1^{n-1} \\ \beta_2^0 & \beta_2^1 & \ldots & \beta_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_r^0 & \beta_r^1 & \ldots u\beta_r^{n-1} & \end{pmatrix}$$

Проверочная матрица H' над $GF(q)$: заменить $\beta_i^j \in GF(q^m)$ на вектора-столбцы длины m из $GF(q)$, соответствующие их разложению по некоторому базису $GF(q^m)$

Пример. код Хемминга: $q = 2, n = 7, g(x) = x^3 + x + 1, \beta_1 = \alpha, \beta_2 = \alpha^2, \beta_3 = \alpha^4$

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha+1 & \alpha^2+\alpha & \alpha^2+\alpha+1 & \alpha+1 \\ 1 & \alpha^2 & \alpha^2+\alpha & \alpha^2+1 & \alpha & \alpha+1 & \alpha^2+\alpha+1 \\ 1 & \alpha^2+\alpha & \alpha & \alpha^2+\alpha+1 & \alpha^2 & \alpha^2+1 & \alpha+1 \end{pmatrix}$$

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

10.2.5 Коды Боуза-Чоудхури-Хоквингема

Определение. Кодом БЧХ над $GF(q)$ длины n с конструктивным расстоянием δ называется циклический код наибольшей возможной размерности, порождающий многочлен которого имеет корни $\alpha^b, \ldots, \alpha^{b+\delta-2}$, где $\alpha \in GF(q^m)$ – примитивный корень степени n из 1

Замечание. В силу теоремы Лагранжа $n|(q^m - 1)$. Если невозможно подобрать такое m соответствующего кода БЧХ не существует

Замечание.

- $n = q^m - 1$ – примитивный код БЧХ
- $b = 1$ – код БЧХ в узком смысле
- $m = 1$ – код Рида-Соломона

10.2.6 Граница БЧХ

Теорема 10.2.3. Если порождающий многочлен циклического кода длины n над $GF(q)$ имеет корни $\alpha^b, \dots, \alpha^{b+\delta-1}$, где $\alpha \in GF(q^m)$ – примитивный корень степени n из 1, то минимальное расстояние этого кода $d \geq \delta$

Доказательство.

- Линейный блоковый код имеет минимальное расстояние d ттогда, когда любые $1, \dots, d-1$ столбцов его проверочной матрицы линейно независимы, но существует d линейно независимых столбцов
- Рассмотрим t $le\delta - 1$ столбцов $j_1, \dots, j_t$ проверочной матрицы над $GF(q^m)$. Первые t ее строк равны

$$\mathcal{H} = \begin{pmatrix} \alpha^{bj_1} & \alpha^{bj_2} & \dots & \alpha^{bj_t} \\ \alpha^{(b+1)j_1} & \alpha^{(b+1)j_2} & \dots & \alpha^{(b+1)j_t} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{(b+t-1)j_1} & \alpha^{(b+t-1)j_2} & \dots & \alpha^{(b+t-1)j_t} \end{pmatrix} =$$

$$= \underbrace{\begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha^{j_1} & \alpha^{j_2} & \dots & a^{j_t} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{(t-1)j_1} & \alpha^{(t-1)j_2} & \dots & \alpha^{(t-1)j_t} \end{pmatrix}}_{W} \operatorname{diag}(a^{bj_1}, \dots, a^{bj_t})$$

- W – матрица Вандермонда, α – примитивный корень степени n из 1 $\implies$ $\alpha^{j_1}, \dots, \alpha^{j_t}$ различны и отличны от 0 $\implies$ W обратима, $\mathcal{H}$ – обратима $\implies$ Любые $le\delta - 1$ столбцов H ЛНЗ над $GF(q^m)$ и $GF(q)$

□

10.2.7 Коды БЧХ

Порождающий многочлен $g(x) = \text{LCM}(M_{\alpha^b}(x), M_{\alpha^{b+1}}(x), \dots, M_{\alpha^{b+\delta-2}}(x))$. Т.к. минимальные многочлены или взаимно просты, или совпадают, порождающий многочлен равен произведению всех различных минимальных многочленов элементов $\alpha^b, \dots, \alpha^{b+\delta-2}$

Замечание. Размерность кода БЧХ $k \geq n - m(\delta - 1)$

- Проверочная матрица над $GF(q^m)$ содержит $\delta - 1$ строк
- Проверочная матрица над $GF(q)$ содержит $m(\delta - 1)$ строк. Некоторые их них могут быть линейно зависимы

Замечание. Двоичные коды БЧХ в узком смысле $(b = 1) : k \geq n - m\lfloor (d-1)/2 \rfloor$

- $M_\beta(x) = M_{\beta^2}(x)$
- $g(x) = \text{LCM}(M_{\alpha^1}(x), M_{\alpha^3}(x), \dots, M_{\alpha^{\delta-2}}(x))$

- В проверочную матрицу над $GF(2^m)$ достаточно включить $\lfloor \frac{d-2}{2} \rfloor$ строк, соответствующих α^{2i+1}

Пример. $(15, 7, 5)$ примитивный код БЧХ в узком смысле над $GF(2)$

- α – примитивный элемент $GF(2^4)$, т.ч. $\alpha^4 + \alpha + 1 = 0$
- $M_\alpha(x) = M_{\alpha^2}(x) = M_{\alpha^4}(x) = x^4 + x + 1$
- $M_{\alpha^3}(x) = x^4 + x^3 + x^2 + x + 1$
- $g(x) = \text{LCM}(M_\alpha(x), M_{\alpha^2}(x), M_{\alpha^3}(x), M_{\alpha^4}(x)) = (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1) = x^8 + x^7 + x^6 + x^4 + 1$

Лекция 11

11.1 Циклические коды

11.1.1 Коды Рида-Соломона

Определение. Код Рида-Соломона – код БЧХ длины $q-1$ над $GF(q)$. Минимальный многочлен $\beta \in GF(q)$ над $GF(q) : M_\beta(x) = x - \beta$. Порождающий многочлен кода Рида-Соломона $g(x) = \prod_{i=0}^{\delta-2}(x-\alpha^{b+i})$. Размерность кода $k = n-\delta+1$. Минимальное расстояние $d \geq \delta$. Граница Синглтона: $d \leq n-k+1 = \delta \implies d = n-k+1$. Код с максимальным достижимым расстоянием

11.1.2 Декодирование кодов БЧХ

Замечание. Рассмотрии исправление ошибок в векторе $y = c + e$.

- $y(x) = a(x)g(x) + e(x)$
- Синдром: $S_i = y(\alpha^{b+i}) = a(\alpha^{b+i})g(\alpha^{b+i}) + e(\alpha^{b+i}) = e(\alpha^{b+i}), 0 \leq i < \delta - 1$
- Пусть ошибки произошли в позициях $j_1, \ldots, j_t, t \leq \lfloor(\delta-1)/2\rfloor$

$$S_i = \sum_{r=0}^{n-1} e_r \alpha^{(b+i)r} = \sum_{l=1}^{t} c_{j_l} \alpha^{(b+i)j_l}$$

- Значение ошибок $E_l = e_{j_l}$
- Локаторы ошибок $X_l = \alpha^{j_l}$

$$S_i = \sum_{l=1}^{t} E_l X_l^{b+i}, 0 \leq i < \delta - 1$$

1. Поиск локаторов ошибок Многочлен локаторов ошибок $\Lambda(x) = \prod_{l=1}^{t}(1-X_l x) = \sum_{l=0}^{t} \Lambda_l x^l$

$$0 = \Lambda(X_i^{-1}) = \sum_{l=0}^{t} \Lambda_l X_i^{-l}, 1 \leq i \leq t$$

$$0 = E_i X_i^{b+j+1} \overbrace{\sum}^{\sum_{l=0}^{t}\Lambda_l X_i^{-l} = \sum_{l=0}^{t}\Lambda_l E_i X_i^{b+j+t-l} =}$$
$$= E_i X_i^{bcj+t} + \Lambda_1 E_i X_i^{b+j+t-l} + \cdots + \Lambda_t E_i X_i^{b+j}, 0 \le j < t$$

$$0 = \sum_{i=1}^{t} E_i X_i^{b+j+t} + \Lambda_1 \sum_{i=1}^{t} E_i X_i^{b+j+t-1} + \cdots + \Lambda_t \sum_{i=1}^{t} E_i X_i^{b+j}$$
$$0 = S_{j+t} + \Lambda_1 S_{j+t-1} + \cdots + \Lambda_t S_j$$

$$\Lambda_1 S_{j+t-1} + \cdots + \Lambda_t S_j = -S_{j+t}$$
$$\underbrace{\begin{pmatrix} S_0 & S_1 & \dots & S_{t-1} \\ S_1 & S_2 & \dots & S_t \\ \vdots & \vdots & \ddots & \vdots \\ S_{t-1} & S_t & \dots & S_{2t-2} \end{pmatrix}}_{\mathbb{S}_t} \begin{pmatrix} \Lambda_t \\ \Lambda_{t-1} \\ \vdots \\ \Lambda_1 \end{pmatrix} = \begin{pmatrix} -S_t \\ -S_{t+1} \\ \vdots \\ -S_{2t-1} \end{pmatrix}$$

Теорема 11.1.1. $\mathbb{S}_z$ обратима, если z равно числу произошедших ошибок t, и вырождена, если $z > t$

Доказательство. $S_i = \sum_{l=1}^{t} E_l X_l^{b+1}., E_z = X_z = 0$ при $z > t$

$$\mathbb{S}_z = \underbrace{\begin{pmatrix} 1 & 1 & \dots & 1 \\ X_1 & X_2 & \dots & X_z \\ \vdots & \vdots & \ddots & \vdots \\ X_1^{z-1} & X_2^{z-1} & \dots & X_z^{z-1} \end{pmatrix}}_{W} \underbrace{\begin{pmatrix} E_1 X_1^b & 0 & \dots & 0 \\ 0 & E_2 X_2^b & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & E_z X_z^b \end{pmatrix}}_{D} \underbrace{\begin{pmatrix} 1 & X_1 & \dots & X_1^{z-1} \\ 1 & X_2 & \dots & X_2^{z-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & X_z & \dots & X_z^{z-1} \end{pmatrix}}_{W^T}$$

D вырождена, если $z > t$ и обратима при $z \le t$. W – матрица Вандермонда, обратима при $z = t$. □

2. Алгоритм Питерсона-Горенстайна-Цирлера

$$\Lambda_1 S_{j+t-1} + \cdots + \Lambda_t S_j = -S_{j+t}$$
$$\underbrace{\begin{pmatrix} S_0 & S_1 & \dots & S_{t-1} \\ S_1 & S_2 & \dots & S_t \\ \vdots & \vdots & \ddots & \vdots \\ S_{t-1} & S_t & \dots & S_{2t-2} \end{pmatrix}}_{\mathbb{S}_t} \begin{pmatrix} \Lambda_t \\ \Lambda_{t-1} \\ \vdots \\ \Lambda_1 \end{pmatrix} = \begin{pmatrix} -S_t \\ -S_{t+1} \\ \vdots \\ -S_{2t-1} \end{pmatrix}$$

- Вычисление синдрома $S_i = y(\alpha^{b+i}), 0 \leq i < \delta - 1$. Сложность при использовании схемы Горнера $O((\delta-1)n)$
- Будем уменьшать предполагаемое число ошибок $t \leq \tau = \lfloor(\delta-1)/2)\rfloor$, пока матрица $\mathbb{S}_t$ не станет обратимой. Проверка обратимости матрицы требует $O(t^3)$ операций
- Решение СЛАУ задает коэффициенты $\Lambda_i, 1 \leq i \leq t$, многочлена локаторов ошибок $\Lambda(x) = 1 + \sum_{i=1}^{t} \Lambda_i x^i$
- Сложность непосредственного подбора t и решения СЛАУ $O(\tau^4)$
- Локаторы ошибок $X_i = \alpha^{j_i} : \Lambda(X_i^{-1}) = 0, 1 \leq i \leq t$. Процедура Ченя поиска корней: подставим в $\Lambda(x)$ все элементы $\alpha^i, 0 \leq i < n$. Сложность $O(nt)$
- Значения ошибок $E_l : S_i \sum_{l=1}^{t} E_l X_l^i, 0 \leq i < t$. Сложность непосредственного решения $O(t^3)$

$$S_i = \sum_{l_1}^{t} E_l X_l^{b+i}, 0 \leq i < \delta - 1 \quad S(x) = \sum_{i=0}^{\delta-2} S_i x^i = \sum_{l=1}^{t} E_l X_l^b \sum_{i=0}^{\delta-2} (X_l x)^i$$

$$\text{1 - } (\mathrm{X}_l x)^{\delta-1} d(1 - X_l x) \left(\sum_{i=0}^{\delta-2} (X_l x)^i\right) = 1 \mod x^{\delta-1}$$
$$\sum_{i=0}^{\delta-2} (X_l x)^i = \frac{1}{1 - X_l x} \mod x^{\delta-1}$$
$$\mathrm{S(x)} = \sum_{l=1}^{t} \frac{E_l X_l^b}{1 - X_l x} \mod x^{\delta-1}$$

Многочлен значений ошибок $\Gamma(x) = \sum_{l=1}^{t} E_l X_l^b \prod_{j \neq l} (1 - X_j x) \equiv \Lambda(x) \sum_{l=1}^{t} \frac{E_l X_l^b}{1 - X_l x} \mod x^{b-1}$.

$$\Gamma(x) \equiv \Lambda(x) S(x) \mod x^{\delta-1}, \deg \Lambda(x) \leq \lfloor(\delta-1)/2\rfloor, \deg \Gamma(x) < \lfloor(\delta-1)/2\rfloor$$

Теорема 11.1.2 (Алгоритм Форни быстрого поиска значений ошибок)**.** $E_i = \frac{X_i^{-b} \Gamma(X_i^{-1})}{\prod_{j \neq i}(1 - X_j X_i^{-1})}, 0 \leq i < t$

Доказательство.

$$\frac{X_i^{-b}(X_i^{-1})}{\prod_{j \neq i}(1 - X_j X_i^{-1})} = \frac{X_i^{-b} \sum_{l=1}^{t} E_l X_l^b \prod_{j \neq i}(1 - X_j X_i^{-1})}{\prod_{j \neq i}(1 - X_j X_i^{-1})} = \frac{E_i \prod_{j \neq i}(1 - X_j X_i^{-1})}{\prod_{j \neq i}(1 - X_j X_i^{-1})} = E_i$$

□

Сложность $O(t^2)$

11.1.3 Расширенный алгоритм Евклида

Поиск наибольшего общего делителя $r_{-1}(x) = a(x), r_0(x) = b(x)$

$$r_{i-1}(x) = q_i(x) r_i(x) + r_{i+1}(x), \deg r_{i+1}(x) < \deg r_i(x)$$

НОД равен последнему ненулевому остатку $r_i(x)$

$$\begin{pmatrix} r_i(x) & r_{i-1}(x) \end{pmatrix} \begin{pmatrix} -q_i(x) & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} r_{i+1}(x) & r_i(x) \end{pmatrix}$$

$$\begin{pmatrix} b(x) & a(x) \end{pmatrix} \underbrace{\prod_i \begin{pmatrix} -q_i(x) & 1 \\ 1 & 0 \end{pmatrix}}_{U(x)} = \begin{pmatrix} 0 & \gcd(a(x), b(x)) \end{pmatrix}$$

Теорема 11.1.3 (Безу)**.** Существуют многочлены $u(x), v(x) : b(x)u(x)+a(x)v(x) = \gcd(a(x), b(x))$

Пусть $U_j(x) = \prod_{i=0}^{j} \begin{pmatrix} -q_i(x) & 1 \\ 1 & 0 \end{pmatrix} = U_{j-1} \underbrace{\begin{pmatrix} -q_j(x) & 1 \\ 1 & 0 \end{pmatrix}}_{Q_j(x)} = \begin{pmatrix} u_{j,0}(x) & u_{j-1,0}(x) \\ u_{j,1}(x) & u_{j-1,1}(x) \end{pmatrix}$, $U_{-1}(x) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

$$\begin{pmatrix} r_0(x) & r_{-1}(x) \end{pmatrix} \begin{pmatrix} u_{j,0}(x) & u_{j-1,0}(x) \\ u_{j,1}(x) & u_{j-1,1}(x) \end{pmatrix} = \begin{pmatrix} r_{j+1}(x) & r_j(x) \end{pmatrix}$$

1. $\deg u_{j,0}(x) = \deg u_{j-1,0}(x) + \deg q_j(x) = \sum_{i=0}^{j} \deg q_i(x) = \sum_{i=0}^{j} (\deg r_{i-1}(x) - \deg r_i(x)) = \deg r_{-1}(x) - \deg r_j(x)$
2. $u_{j,0}(x)u_{j-1,1}(x) - u_{j-1,0}(x)u_{j,1}(x) = \det U_j(x) = \prod_{i=0}^{j} \det Q_j(x) = (-1)^{j+1}$
3. $\gcd(u_{j,0}(x), u_{j,1}(x))1$. Если $f(x)|u_{j,0}(x), f(x)|u_{j,1}(x)$, то $f(x)|(u_{j,0}(x)u_{j-1,1}(x)-u_{j-1,0}(x)u_{j,1}(x))$
4. $r_{j+1}(x) = r_0(x)u_{j,0}(x) + r_{-1}(x)u_{j,1}(x)$
   $r_{j+1}(x) \equiv r_0(x)u_{j,0}(x) \mod r_{-1}(x)$ – похоже на ключевое уравнение
5. $\gcd(r_{j+1}(x), u_{j,0}(x)) = \gcd(r_{-1}(x), u_{j,0}(x))$
   $f(x)|r_{j+1}(x) \wedge f(x)|usj, 0(x) \implies f(x)|r_{-1}(x)$ $f(x)|r_{-1}(x) \wedge f(xj|u_{j,0}(x) \implies f(x)|r_{j+1}(x)$

11.1.4 Алгоритм Сугиямы

Пусть $\delta = 2\tau + 1$

1. Пусть $r_{-1}(x) = x^{2\tau}, r_0(x) = S(x)$
2. Выполнять $r_{i-1}(x) = q_i(x)r_i(x) + r_{i+1}(x)$, пока не получится $\deg r_i(x) \geq \tau, \deg r_{i+1}(x) < \tau$
3. $\Gamma(x) = r_{i+1}(x), \Lambda(x) = u_{i,0}(x)$

Корректность алгоритма

1. Степени $r_i(x)$ монотонно убывают, т.е. условие останова достижимо
2. $\Gamma(x) = r_{i+1}(x) = r_0(x)u_{i,0}(x) + r_{-1}(x)u_{i,1}(x) = S(x)\Lambda(x) + x^{2\tau}u_{i,1}(x) \equiv S(x)\Lambda(x) \mod x^{2\tau}$
3. $\deg u_{i,0}(x) = \deg r_{-1}(x) - \deg r_i(x) \leq 2\tau - \tau \leq \tau$
4. Пусть $\Gamma'(x) \equiv S(x)\Lambda'(x) \mod x^{2\tau}, \deg \Lambda'(x) \leq \tau, \deg \Gamma'(x) < \tau$. Если $\Lambda'(x), \Gamma'(x)$ – истинные многочлены локаторов и значений ошибок, то $\gcd(\Lambda'(x), \Gamma'(x)) = 1$

   $$\Gamma'(x)\Lambda(x) \equiv \Lambda(x)S(x)\Lambda'(x) \equiv \Gamma(x)\Lambda'(x) \mod x^{2\tau}$$

   $\deg \Gamma'(x)+\deg \Lambda(x) < 2\tau, \deg \Gamma(x)+\deg \Lambda'(x) < 2\tau \implies \Gamma'(x)\Lambda(x) = \Gamma(x)\Lambda'(x)$ Из взаимной простоты $\Lambda'(x), \Gamma'(x)$ следует, что $\mu(x) = \frac{\Lambda(x)}{\Lambda'(x)} = \frac{\Gamma(x)}{\Gamma'(x)}$ – многочленю

   $$\Gamma'(x)\mu(x) = \Gamma(x) = S(x)\Lambda(x) + x^{2\tau}u_{i,1}(x) = S(x)\Lambda'(x)\mu(x) + x^{2\tau}u_{i,1}(x)$$

   $\implies \mu(x)|u_{i,1}(x)$. Но $\Lambda(x) = \mu(x)\Lambda'(x) = u_{i,0}(x)$ и $u_{i,1}(x)$ взаимно просты $\implies \mu(x) = const$

11.1.5 Сложность декодирования кодов БЧХ и Рида-Соломона

- Вычисление синдрома
  - Схема Горнера: $S_i = y(\alpha^{b+i}) = y_0 + \alpha^{b+i}(y_1 + \alpha^{b+i}(y_3 + \dots))), 0 \le i < \delta$. Сложность $O(n\delta)$ операций
  - Метод Герцеля: $r_i(x) \equiv y(x) \mod M_{\alpha^{b+i}}(x)$; $S_i = r_i(\alpha_{b+i}), \alpha \in GF(p^m)$. $M_{\alpha^{b+i}} \in GF(p)[x]$ – минимальный многочлен α^{b+i}. Деление на него тербует только сложений. Минимальные многочлены многих α^{b+i} совпадают
- Решение ключевого уравнения $\Gamma(x) \equiv S(x)\Lambda(x) \mod x^{\delta-1}$. Расширенный алгоритм Евклида: $O(\delta^2)$ операций
- Поиск корней X_i^{-1} многочлена локаторов ошибок $\Lambda(x)$. Процедура Ченя (перебор $\alpha^i, 0 \le i < n$ и проверка $\Lambda(\alpha^i) = 0$) со сложностью $O(n\delta/2)$
- Поиск значений ошибок. Метод Форни со сложностью $O(\delta^2)$

Лекция 12

12.1 Декодирование БЧХ

- Дана последовательность $S_0, \ldots, S_{\delta-2}$
- Как восстановить регист сдвига минмальной длины, порождающий эту последовательность по ее начальной части?

$$\Lambda_1 S_{j+t-1} + \cdots + \Lambda_t S_j d - S_{j+1}$$

- Фильтр $(L, \Lambda^{(n)}(x) = 1 + \sum_{i=1}^{L} \Lambda_i^{(n)} x^i)$ порождает последовательность S_0^{n-1}, если

$$S_k = -\sum_{i=1}^{L} \Lambda_i^{(n)} S_{k-i}, L \le k \le n-1$$

  Параметры L и $\Lambda^{(n)}(x)$ назваются длиной фильтра (РСЛОС) и многочленом связей

- В общем случае $\deg \Lambda^{(n)}(x) \le L$

Доделать Картинка

12.1.1 Минимальный РСЛОС

Лемма 2. *Пусть фильтры $(L_{n-1}, \Lambda^{(n-1)}(x))$ и $(L_n, \Lambda^{(n-1)}(x))$ порождает последовательности S_0^{n-2} и S_0^{n-1} соответственно, причем $(L_{n-1}, \Lambda^{(n-1)}(x))$ не прождает $S_0^{(n-1)}$, и величины L_{n-1} и L_n являются наименьшими возможными. Тогда $L_n \ge \max(L_{n-1}, n - L_{n-1})$*

Доказательство. Фильтр, порождающий S_0^{n-1}, обязан порождать и S_0^{n-2}, поэтому $L_n \ge L_{n-1}$. Покажем, что если фильтр $(L_{n-1}, \Lambda^{(n-1)}(x))$ порождает S_0^{n-2}, но не порождает S_0^{n-1}, то $L_n \ge n - L_{n-1}$. Предположим, что это не так. Тогда

$$S_{n-1} \ne -\sum_{i=1}^{L_{n-1}} \Lambda_i^{(n-1)} S_{n-1-i} = \sum_{i=1}^{L_{n-1}} \Lambda_i^{(n-1)} \sum_{k=1}^{L_n} \Lambda_k^{(n)} S_{n-1-k-i}$$

Полседний переход возможен в силу того, что для всех i выполняется $L_n \le n - L_{n-1} - 1 \le n - i - 1 \le n - 2$, т.е. величины S_{n-1-i} могут быть порождены с помощью $(L_n, \Lambda^{(n)}(x))$. Меняя порядок суммирования, получим

$$S_{n-1} \ne \sum_{k=1}^{L_n} \Lambda_k^{(n)} \sum_{i=1}^{L_{n-1}} \Lambda_i^{(n-2)} S_{n-1-k-i} = -\sum_{k=1}^{L_n} \Lambda_k^{(n)} S_{n-1-k} = S_{n-1}$$

Из полученного противоречия вытекает , что $L_n \geq n - L_{n-1}$ □

Теорема 12.1.1. Предположим, что РСЛОС $(L_i, \Lambda^{(i)}(x))$ порождает $S_0^i, 0 \leq i \leq r-1$, причем величины L_i являются нименьшими возможными. Тогда РСЛОС с многочлено связей

$$\Lambda^{(r)}(x) = \begin{cases} \Lambda^{(r-1)}(x) & \text{,если } \Delta_r^{(r)} = 0 \\ \Lambda^{(r-1)}(x) - \frac{\Delta_r^{(r)}}{\Delta_m^{(m)}} x^{r-m} \Lambda^{(m-1)}(x) & \text{,если } \Delta_r^{(r)} = 0 \end{cases} \tag{12.1}$$

порождает S_0^{r-1} и имеет нименьшую длину. Здесь $\Delta_r^{(v)} = \sum_{j=0}^{L_{r-1}} \Lambda_j^{(v)} S_{r-1-j}$ – невязка, m – наибольшее число, меньшее r, такое, что $L_{m-1} < L_m$, и $\Delta_0^{(0)} = 1$

- $\Delta_r^{(v)}$ равна разности истинного значения S_{r-1} и значения, вычисленно с помощью $(L_v, \Lambda^{(v)}(x))$
- Наименьшая возможная длина РСЛОС $L_i = \max(L_{i-1}, i - L_{i-1})$

Доказательство. Будем считать, что при $r = 0 \Lambda^{(0)}(x) = 1$, и покажем, что на каждом шаге правило 12.1 приводит к РСЛОС наименьшей длины. При $\Delta_r^{(r)} = 0$ в изменении РСЛОС нет необходимости, т.е. $\Lambda^{(r)}(x) = \Lambda^{(r-1)}(x)$. Если старый РСЛОС имел наименьшую длину, то он сохраняет это свойство и для S_0^{r-1}. В противном случае: Длины РСЛОС менялась на шаге $m \implies L_{r-1} = m - L_{m-1}, L_{m-1} < L_{r-1}$ Модифицированный многочлен имеет степень /(° $\Lambda^{(r)}$ (x) $\leq$ max(° $\Lambda^{(r-1)}$ (x), r - m + ° $\Lambda^{(m-1)}$(x)) $\leq$ max(L_{r-1}, r - m + L_{m-1}) = max(L_{r-1}, r - L_{r-1}) $\implies$ $\Lambda^{(r)}(x)\backslash$) может рассматриваться как многочлен связей для РСЛОС с оптимальной длиной $L_r = \max(L_{r-1}, r - L_{r-1})$. Невязка, соответствующая модифицированному РСЛОС, равна

$$\Delta_r^{(r)} = \sum_{j=0}^{L_{r-1}} \Lambda_j^{(r-1)} S_{r-1-j} - \frac{\Delta_r^{(r-1)}}{\Delta_m^{(m-1)}} \sum_{j=0}^{L_{m-1}} \Lambda_j^{(m-1)} S_{r-1-j-(r-m)} = \Delta_r^{(r-1)} - \frac{\Delta_r^{(r-1)}}{\Delta_m^{(m-1)}} \Delta_m^{(m-1)} = 0$$

Таким образом, $(L_r, \Lambda^{(r)}(x))$ действительно порождает S_{r-1}. $(L_r, \Lambda^{(r)}(x))$ порождает и предшествующие элементы S_{k-1}

$$\Delta_k^{(r)} = \sum_{j=0}^{L_r} \Lambda_j^{(r)} S_{k-1-j} = \sum_{j=0}^{L_{r-1}} \Lambda_j^{(r-1)} S_{k-1-j} - \frac{\Delta_r^{(r-1)}}{\Delta_m^{(m-1)}} \sum_{j=0}^{L_{m-1}} \Lambda_j^{(m-1)} S_{k-1-j-(r-m)} =$$

$$= \Delta_k^{(r-1)} - \frac{\Delta_r^{(r-1)}}{\Delta_m^{(m-1)}} \Delta_{k-r+m}^{(m-1)} = 0, L_{r-1} + 1 \leq L_r + 1 \leq k \leq r - 1$$

Последнее равенство вытекает их того, что РСЛОС $(L_{r-1}, \Lambda^{(r-1)}(x))$ порождает S_0^{r-2}, т.е. $\Delta_k^{(r-1)} = 0, L_{r-1}+1 \leq k \leq r-1$, и $(L_{m-1}, \Lambda^{(m-1)}(x))$ порождает S_0^{m-2}, т.е. $\Delta_k^{(m-1)} = 0, L_{m-1}+1 \leq k \leq m-1$ □

12.1.2 Алгоритм Берлекэмпа-Месси

Program 8 Алгоритм Берлекэмпа-Месси

1: $\Lambda(x) \leftarrow 1, r \leftarrow 1, m \leftarrow 0, L \leftarrow 0, B(x) \leftarrow 1$
2: **while** $\Delta_r \leftarrow \sum_{j=0}^{L} \Lambda_j S_{r-1-j}$ **do**
3: **if** $\Delta_r \neq 0$ **then**
4: $T(x) \leftarrow \Lambda(x) - \Delta_r x^{r-m} B(x)$
5: **if** $2L \leq r-1$ **then**
6: $B(x) \leftarrow \Delta_r^{-1} \Lambda(x)$
7: $\Lambda(x) \leftarrow T(x)$
8: $L \leftarrow r - L$
9: $m \leftarrow r$
10: **else**
11: $\Lambda(x) \leftarrow T(x)$
12: **end if**
13: **end if**
14: $r \leftarrow r + 1$
15: **end while**
16: **return** $(L, \Lambda(x))$

- Если $L \neq \deg \Lambda(x)$, число ошибок превышает $(\delta - 1)/2$ (крому случая расширенных кодом БЧХ)
- Сложность $O((\delta-1)^2)$
- Самый простой этап декодирования кодов БЧХ
- Для двоичных кодов БЧХ в укзом смысле $\Delta_{2i} = 0 \implies$ половину итераций можно пропустить

12.2 Мягкое декодирование кодов БЧХ

12.2.1 Метод Чейза-2 мягкого декодирования

- Найдем τ наименее надежных символов принятого векотора $(y_0, \ldots, y_{n-1})$. Пусть они расположены в позициях $0, \ldots, \tau - 1$
- Пусть $\hat{y}_i \in GF(q)$ – жесткое решение относительно y_i
- Переберем все q^τ векторов $(x_0, \ldots, x_{\tau-1}, \hat{y}_\tau, \hat{y}_{\tau+1}, \ldots, \hat{y}_{n-1}), x_i \in GF(q)$. Для кжадого такого вектора выполним его жесткое декодирование с исправлением $(\delta - 1)/2$ ошибок
- Из полученных кодовых слов выберем наиболее вероятное для $(y_0, \ldots, y_{n-1})$
- Сложность $O(2^r \delta^2)$

12.2.2 Метод Пинди декодирвания с мягким выходом

- Апостериорные ЛОПП

$$L_i = \ln \frac{\sum_{c\in\mathcal{C}:c_i=0} \prod_{j=0}^{n-1} P(c_j|y_j)}{\sum_{c\in\mathcal{C}:c_i=1} \prod_{j=0}^{n-1} P(c_j|y_j)} \approx \ln \frac{\max_{c\in\mathcal{C}:c_i=0} \prod_{j=0}^{n-1} P(c_j|y_j)}{\max_{c\in\mathcal{C}:c_i=0} \prod_{j=0}^{n-1} P(c_j|y_j)} = \min_{c\in\mathcal{C}:c_i=1} E(c,y) - \min_{c\in\mathcal{C}:c_i=0} E(c,y)$$

- Пусть $\mathcal{L}$ – список, полученный декодированием y в коде $\mathcal{C}$ (Чейз-2б Тал-Варди, ...), и $\hat{c}$ – наиболее вероятный его элемент
- $L_i \approx \min_{c\in\mathcal{L}:c_i=1} E(c,y) - \min_{c\in\mathcal{L}:c_i} = 0E(c,y)$
- Внешние ЛОПП:

$$\hat{L}_i = \begin{cases} L_i - \ln \frac{P(y_i|c_i=0)}{P(y_i|c_i=1)} & \text{, если } \exists c', c'' \in \mathcal{L} : c'_i = 0, c''_i = 1 \\ \beta(1-2\hat{c}_i) & \text{, иначе} \end{cases}$$

, где β – экспериментально подбираемый параметр

12.3 QR-коды (1967)

- Число y называется квадратичным вычетом *quadraticresidue* по модулю n, если существует решени сравнения $x^2 \equiv y \mod n$. В противном случае y – квадратичный невычет
- $(n-x)^2 \equiv x^2 \mod n \implies$ квадратичными вычетами являются $1^2, 2^2, \ldots, ((n-1)/2)^2 \mod n$
- Произведение квадратичных вычетом – квадратичный вычет

Определение. Квадратично-вычетным называется циклический код длины n над полем $GF(p)$ с порождающим многочленом $g_1(x) = \prod_{i\in Q}(x-\alpha^i), (x-1)g_1(x), g_2(x) = \prod_{i\in N}(x-\alpha^i)$ или $(x-1)g_2(x)$, где n – простое число, p – квадратичный вычет по модулю n, $\alpha \in GF(p^m)$ – примитивный корень степени n из 1, Q и N – множество квадратичных вычетов и невчетов по модулю n

Минимальное расстояние d КВ кода удовлетворяет $d^2 \geq n$. Если $n = 4s-1, s \in \mathbb{N}$, то $d^2 - d + 1 \geq n$

12.4 QR-коды (1994) (жалкая подделка)

- Quick Response code
- Данные представляются в виде черных и белых точек
- Для защиты от ошибок считывания используются коды Рида-Соломона с различными параметрами
- Возможность исправления ошибок позволяет создавать художественные QR-коды

12.5 Cyclic Redundancy Check

- CRC – циклический код, используемый для обнаружнеия ошибок
- Контрольная сумма (многочлен проверочных символов $r(x)$) для многочлена данных $a(x)$ вычисляется с помощью формулы систематического кодирования $r(x) \equiv x^{N-K}a(x) \mod g(x), \deg a(x) \leq K-1$
- Число проверочных символов равно $N-K = \deg g(x), N \leq n$

$$g(x)|(x^n-1)$$

- $K \leq n - \deg g(x)$
- Нельзя допускать $K > n - \deg g(x)$, т.к. это приведем к коду с неизвестным (вероятно, плохим) минимальным расстоянием

12.6 Выводы

- Циклические коды допускают еще более компактное задание по сравнению с линейными блоковыми кодами
- Конструкция кодов БЧХ позволяет получить коды с заданным минимальным расстоянием
- Коды Рида-Соломона – коды БЧХ, лежащие на границе Синглтона
- Существуют алгоритмы декодирования кодов БЧХ с исправлением $\lfloor(\delta-1)/2\rfloor$ ошибок со сложностью $O(n\delta+\delta^2)$
- Расширенные примитивные коды БЧХ в узком смысле достигают предела Шеннона для двоичного стирающего канала
- Метод Чейза-Пиндии мягкого декодирования

Лекция 13

13.1 Альтернантные коды

Теорема 13.1.1. $= (_0, \ldots, c_{n-1})$ – кодовое слово кода Рида-Соломона над $GF(q)$ в узком смысле ттогда, когда $c_i = f(\alpha_i), 0 \le i < n$ (т.е. $c = ev(f)$), где $\deg f(x) < k, f(x) \in GF(q)[x]$

Доказательство. Доделать □

Определение. $n, k, n-k+1$ кодом Рида-Соломона называется множество векторов $c = (c_0, \ldots, c_{n-1})$, где $c_i = f(a_i), \deg f(x) < k, f(x) \in GF(q)[x], a_i \in GF(q)$ – различные значения (локаторы)

Определение. Обобщенным $(n, k, d = n-k+1)$ кодом Рида-Соломона $GRS(n, k, a, u)$ называется множество векторов $(c_0u_0, \ldots, c_{n-1}u_{n-1})$, где $(c_0, \ldots, c_{n-1})$ – кодовое слово $(n, k, n-k+1)$ кода Рида-Соломона (т.е. $c_i = f(a_i), \deg f(x) < k, a_i$ – различные), и $u_0, \ldots, u_{n-1}$ – ненулевые константы

Определение. Альтернантным кодом длины n над полем $GF(q)$ называется код $\mathcal{A}(n, r, a, u)$ с проверочной матрицей

$$H = \begin{pmatrix} a_0^0 & a_1^0 & \ldots & a_{n-1}^0 \\ a_0^1 & a_1^1 & \ldots & a_{n-1}^1 \\ \vdots & \vdots & \ddots & \vdots \\ a_0^{r-1} & a_1^{r-1} & \ldots & a_{n-1}^{r-1} \end{pmatrix} (u_0, u_1, \ldots, u_{n-1}) = (H_{i,j})$$

где $a_i \in GF(q^m)$ – различные элементы, $u_i \in GF(q^m) \setminus 0$

Замечание. Доделать

- Минимальное расстояние $d \ge r + 1$
- Размерность $n - r \ge k \ge n - mr$

Теорема 13.1.2. Пусть $m|(n-h)$. Существует альтернантный $(n, k \ge h, d \ge \delta)$ код над $GF(q)$ такой, что

$$\sum_{i=1}^{\delta-1} (q-1)^i C_n^i < (q^m - 1)^{\frac{n-h}{m}}$$

Замечание.

- Рассмотрим $\mathcal{A}(n,(n-h)/m,a,u) = GRS(n,n-(n-h)/m,a,v) \cap GF(q)^n$

Доделать

Общее количество альтернантных кодом больше чем количество плохих альтернантных кодов, значит есть хорошие альтернантные коды

Замечание.

$$\sum_{i=0}^{d-2} C_{n-1}^i (q-1)^i < \underbrace{\sum_{i=0}^{d-1} C_n^i (q-1)^i < (q^m-1)^{\frac{n-h}{m}}}_{\text{Альтернантные коды}} < q^{n-h}$$

13.1.1 Коды Гоппы

Определение. Пусть задан многочлен (многочлен Гоппы) $G(x) \in GF(q^m)[x]$ и $a_0,\dots,a_{n-1} \in GF(q^m)$, причем $G(a_i) \neq 0$. Кодом Гоппы называется множество $(c_0,\dots,c_{n-1}) \in GF(q)^n$.

$$\sum_{i=0}^{n-1} \frac{c_i}{x-a_i} \equiv 0 \mod G(x)$$

Утверждение. *Коды Гоппы являются альтернантными*

Доказательство. Доделать □

Замечание. Двоичные коды Гоппы Доделать

13.1.2 Криптосистема Мак-Элиса

Доделать

Лекция 14

14.1 Кодовая модуляция

Замечание. Пропускная способность канала без памяти с входным алфавитом $\mathcal{X}$ и выходным алфавитом $\mathcal{Y}$

$$C = \max_{p_X} \int_{\mathcal{Y}} \int_{\mathcal{X}} p_{XY}(x,y) \log_2 \frac{p_{XY}(x,y)}{p_X(x) p_Y(y)} dxdy$$

Доделать

Лекция 15

15.1 Сетевое кодирование

Доделать