НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Модуль 1. Математические основы помехоустойчивого кодирования

Постановка задачи кодирования. Основные теоремы и базовые понятия. Код с повторением и проверкой на четность, код Хэмминга.

Иванов Ф. И.
к.ф.-м.н., доцент

Национальный исследовательский университет
«Высшая школа экономики»

Основная задача

Основная задача кодирования - защита передаваемой информации от шумов.

Ни одна система связи не обходится без кодирования.

Без кодирования невозможны:

- Беспроводная связь (3G, LTE, Wi-Fi);
- Проводная связь (ВОЛС, Ethernet);
- Системы хранения данных (CD, DVD, HDD, SSD и т.д.).
- Даже QR-коды построены на принципах теории кодирования

Модель передачи данных

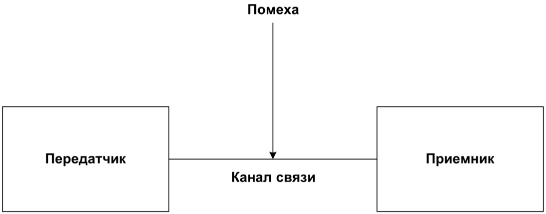


Рис.: Простейшая модель передачи данных

- Передатчик передает последовательности длины n из 0 и 1: $\mathbf{u} = (u_1, u_2, \ldots, u_n)$, $u_i \in \{0,1\}$.
- В канале действует случайная помеха: каждый символ передаваемой последовательности независимо от других может быть искажен с вероятностью $\tau < 1/2$.

Двоичный симметричный канал

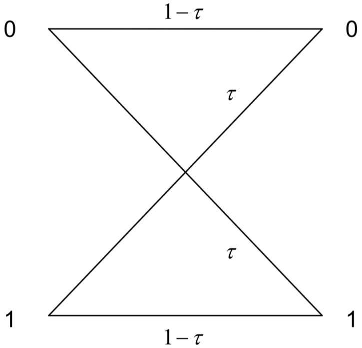


Рис.: Двоичный симметричный канал

$$p(0|0) = p(1|1) = 1 - \tau,\ p(0|1) = p(1|0) = \tau$$

Простейшая модель ошибки

Введем в рассмотрение вектор:

$$\mathbf{e} = (e_1, e_2, \ldots, e_n), u_i \in \{0{,}1\},$$

причем $e_i = 1$ тогда и только тогда, когда произошло искажение в i символе $\mathbf{u}$.

Пример

Пусть $\mathbf{u} = (110001001)$*, а на приемном конце получен вектор* $\mathbf{v} = (010001101)$*. Это означает, что в канале подверглись искажению первый и седьмой символы передававшегося вектора. Это означает, что в векторе* $\mathbf{e}$ *единицы расположены на первом и седьмом местах, т.е.* $\mathbf{e} = (100000100)$*.*

Легко заметить, что

$$\mathbf{v} = \mathbf{u} + \mathbf{e},$$

где $0 + 0 = 1 + 1 = 0$, $0 + 1 = 1 + 0 = 1$

Простейший код - код с повторением

Пусть через канал передаются только 2 возможных сообщения: 0 или 1. Выберем произвольное $n > 1$ и будем осуществлять кодирование сообщений следующим образом:

$$0 \to (0, \dots 0); 1 \to (1, \dots 1),$$

т. е. дублируем символ n раз.
Так как $\tau < 1/2$, то $n\tau < n/2$ и при $n \to \infty$

$$P_e = P(k > n/2) = \sum_{i=n/2+1}^{n} \binom{n}{i} \tau^i (1-\tau)^{n-i} \to 0$$

Но вместо 1 бита передаем n:

$$R = 1/n$$

Основная идея кодирования

Вероятность передать k некодированнных бит без ошибок:

$$P(k) = (1 - \tau)^k,$$

Например, при $\tau = 10^{-3}$, $P(1000) < 0.37$, а $P(10000) < 10^{-4}$

Основная идея – добавление избыточности

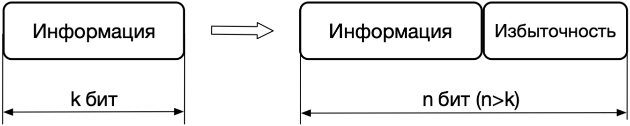
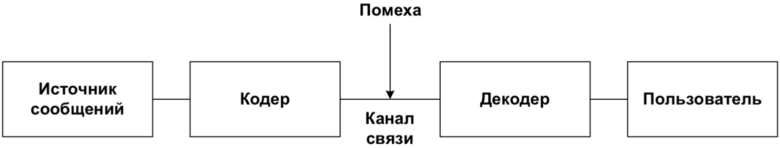


Рис.: Основная идея кодирования

Основные термины

- **Код** - произвольное множество векторов длины n: $C = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_M\}$, $\mathbf{c}_i = (c_i^1, c_i^2, ..., c_i^n)$, $c_i^j \in \{0,1\}$. Обычно код обозначают как (n,k) или (n,M).
- **Информационный вектор** — произвольный двоичный вектор длины k, $k < n$.
- n — **длина кода**, M — **мощность кода**, $R = k/n$ или $R = \log_2 M/n$ — **скорость кода**
- **Кодирование** — преобразование $\phi()$: $\phi(\mathbf{u}) = \mathbf{c} \in C$, где $\mathbf{u}$ — информационный вектор.
- **Декодирование** — преобразование $\phi^{-1}()$: $\phi^{-1}(\mathbf{v}) = \widetilde{\mathbf{u}}$, где $\widetilde{\mathbf{u}}$ — оценка информационного вектора, а $\mathbf{v}$ — принятое (возможно с ошибками) слово

Расширенная модель передачи



1. Информационный вектор $\mathbf{u}$ длины k поступает из источника в кодер
2. Кодер ϕ вычисляет кодовое слово $\mathbf{c} = \phi(\mathbf{u})$ длины $n > k$
3. Кодовое слово передается через канал, где вносятся ошибки. В итоге приемник принимает $\mathbf{v} = \mathbf{c} + \mathbf{e}$
4. Декодер пытается восстановить информацию из принятого вектора $\phi^{-1}(\mathbf{v}) = \widetilde{\mathbf{u}}$
5. Получатель принимает $\widetilde{\mathbf{u}}$ или уведомление об отказе декодирования

Основная теорема кодирования

Теорема

(Прямая теорема Шеннона) Существует (n,M) *код такой, что если* $N \to \infty$ *и его скорость* $R < C$, *то* $\forall \epsilon > 0$ *вероятность ошибки декодирования* $P_e < \epsilon$.

$C = 1 - h(\tau)$ — пропускная способность ДСК,
$h(x) = -x * \log_2(x) - (1-x) * \log_2(1-x)$ — функция двоичной энтропии.

Минимальное расстояние

На множестве V^n всех двоичных векторов длины n введем метрику (Хэмминга):

$$d(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i|$$

Например, если $\mathbf{x} = (11011)$, $\mathbf{y} = (01010)$, тогда $d(\mathbf{x},\mathbf{y}) = 2$.
Для кода C определим величину минимального расстояния Хэмминга:

$$d_{min}(C) = \min_{\mathbf{x},\mathbf{y} \in C, \mathbf{x} \neq \mathbf{y}} d(\mathbf{x},\mathbf{y})$$

Минимальное расстояние и исправление ошибок

Пусть $d_{min}(C) = 2t + 1$:

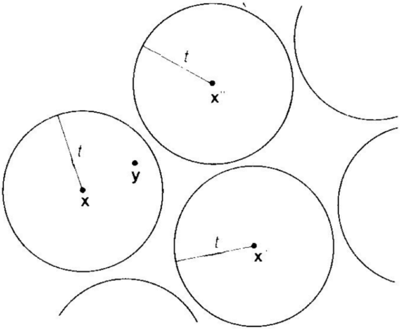


Рис.: Сферы вокруг слов

Минимальное расстояние и исправление ошибок

- Сферы радиуса t такие, что их центры - это кодовые слова, не пересекаются
- Если вместо переданного слова $\mathbf{x} \in C$ было принято $\mathbf{y} \in V^n$, причем $d(x,y) \leq t$, то $\mathbf{y}$ "не выскочит"из сферы радиуса t, т. е. $\forall \mathbf{x}' \in C, \mathbf{x}' \neq \mathbf{x}: \; d(\mathbf{x},\mathbf{y}) < d(\mathbf{x}',\mathbf{y})$.
- При переборе всех кодовых слов ближайшим к $\mathbf{y}$ окажется переданное $\mathbf{x}$, т. е. декодирование будет успешным

Теорема

При $d \geq 2t + 1$ код исправляет все ошибки кратности до t.

Минимальное расстояние и исправление ошибок

Пусть $d = 2t + 2$, тогда:

- Сферы радиуса $t + 1$ такие, что их центры - это кодовые слова, не пересекаются, но могут касаться
- Если вместо переданного слова $\mathbf{x} \in C$ было принято $\mathbf{y} \in V^n$, причем $d(x,y) \leq t$, то $\mathbf{y}$ "не выскочит"из сферы радиуса t, т. е. $\forall \mathbf{x}' \in C, \mathbf{x}' \neq \mathbf{x} : \ d(\mathbf{x},\mathbf{y}) < d(\mathbf{x}',\mathbf{y})$ - ошибка будет исправлена
- Если же $d(x,y) = t + 1$, то может найтись еще одно слово $\mathbf{x}' \in C, \mathbf{x}' \neq \mathbf{x} : \ d(\mathbf{x},\mathbf{y}) = d(\mathbf{x}',\mathbf{y})$, то есть при декодировании обнаружится по крайней 2 слова, равноудаленных от принятого - ошибка обнаружена.

Теорема

При $d \geq 2t + 2$ код исправляет все ошибки кратности до t и обнаруживает ошибки кратности $t + 1$.

Минимальное расстояние и обнаружение ошибок

Пусть $d = t + 1$, тогда:

- Сферы радиуса t такие, что их центры - это кодовые слова, могут пересекаться, но каждая сфера включает единственное кодовое слово
- Если вместо переданного слова $\mathbf{x} \in C$ было принято $\mathbf{y} \in V^n$, причем $d(x,y) \leq t$, то $\mathbf{y}$ "не перейдет"ни в какое другое кодовое слово (так как слово в сфере радиуса t единственное) - ошибка обнаружена.

Теорема

При $d \geq t + 1$ код обнаруживает все ошибки кратности до t.

Параметры простейших кодов - код с повторением

Пример

Код с повторением R: количество информационных символов $k = 1$, *длина* n, *скорость* $R = 1/n$, *код состоит из 2-х слов:* $(0, 0, ..., 0)$ *и* $(1, 1, ..., 1)$. *Минимальное расстояние равно* $d = n$, *число исправляемых ошибок:* $t = \frac{n-1}{2}$.

Данный код исправляет ошибки до половины своей длины и является оптимальным: не существует кода с большим минимальным расстоянием. Основной недостаток - очень низкая скорость.

Параметры простейших кодов - код с проверкой на четность

Пример

Код с проверкой на четность P: количество информационных символов $k = n - 1$, длина n, скорость $R = \frac{n-1}{n}$, код состоит из 2^{n-1} слова: $\mathbf{c} = (c_1, c_2, ..., c_{n-1}, p_n)$, где $c_1, c_2, ..., c_{n-1}$ - произвольные биты (информационные символы), а

$$p_n = \sum_{k=1}^{n-1} c_k$$

Все слова кода имеют четный вес. Минимальное расстояние равно $d = 2$, то есть код обнаруживает любую одиночную (нечетной кратности) ошибку.

Данный код обнаруживает все одиночные и является оптимальным: не существует кода с большим минимальным расстоянием при данном числе информационных символов.

Код Хэмминга

Рассмотрим матрицу

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Построим код H как множество векторов, образующих ядро этой матрицы:

$$H = \{\mathbf{c} \in V^n : \mathbf{c}\mathbf{H}^T = \mathbf{0}\}.$$

По сути это означает, что если $(c_1, c_2, \ldots, c_7) \in H$, то:

$$c_2 + c_3 + c_4 + c_5 = 0$$

$$c_1 + c_2 + c_3 + c_6 = 0$$

$$c_1 + c_2 + c_4 + c_7 = 0$$

Декодирование кодов Хэмминга

Рассмотрим $\mathbf{c} \in H$: $\mathbf{c} = (1000011)$.
Внесем одну ошибку на 4 позицию и примем:
$(1000011) + (0001000) = (1001011) = \mathbf{v}$.
Вычислим:

$$\mathbf{vH}^T = (1001011) \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}^T = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

Это 4 столбец матрицы $\mathbf{H}$ — ошибка исправлена!

Параметры кодов Хэмминга

В общем случае код Хэмминга H_m имеет параметры:

- Длина $n = 2^m - 1$
- Число информационных символов $k = 2^m - m - 1$
- Минимальное расстояние $d = 3$ — исправляет одну ошибку
- Матрица $\mathbf{H}$, определяющая код Хэмминга состоит из всех $2^m - 1$ различных ненулевых столбцов высоты m
- Код Хэмминга — совершенный. Это плотная упаковка пространства V^n: все пространство распадается на содержимое сфер радиуса 1 с центрами в кодовых словах кода H_m.

Границы в теории кодирования

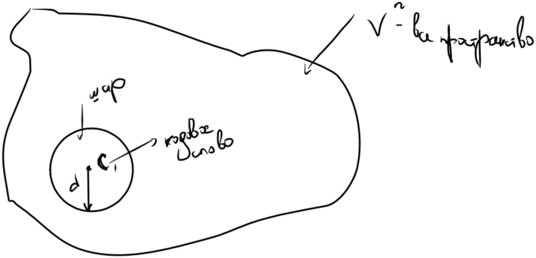
Нам бы хотелось построить код C у которого при заданной длине n:

1. Максимальное d — наилучшая корректирующая способность
2. Максимальная R — наибольшая скорость передачи
3. Простые процедуры кодирования и декодирования

Но пункты (1)-(2) противоречат друг другу!
У нас есть $(n, 1, n)$ код с повторением и $(n, n-1, 2)$ код с проверкой на четность!
Увеличивая расстояние d мы уменьшаем R и наоборот!
Вероятно, есть границы $\delta(n, d, R)$ связывающая эти величины...

Граница Гилберта

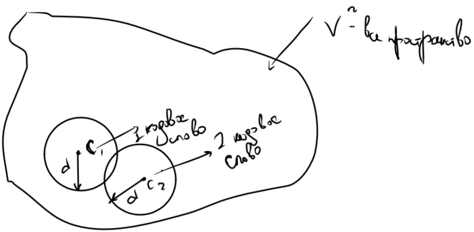
Будем строить (n,M,d) код слово за слово. Также будем считать, что число информационных символов $k = \log_2 M$.



Первое слово $\mathbf{c}_1$ берем любым из 2^n доступных. Выбрасываем "шар" — это те слова, которые находятся на расстоянии меньше чем d от выбранного $\mathbf{c}_1$. Таких слов будет $\sum_{i=1}^{d-1} \binom{n}{i}$.

Граница Гилберта

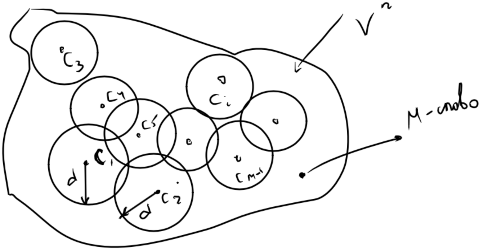
У нас есть 1 слово $\mathbf{c}_1$ и $2^n - \sum\limits_{i=1}^{d-1} \binom{n}{i}$ способов выбрать второе слово. Все эти слова заведомо находятся на расстоянии по крайней мере d от $\mathbf{c}_1$. Второе слово $\mathbf{c}_2$ выбираем случайно.



Снова выбрасываем "шар" — это те слова, которые находятся на расстоянии меньше чем d от выбранного $\mathbf{c}_2$. Таких слов будет $\sum\limits_{i=1}^{d-1} \binom{n}{i}$. Тогда уже выбросили не более чем $2\sum\limits_{i=1}^{d-1} \binom{n}{i}$ слов.

Граница Гилберта

Продолжаем этот процесс выбрасывания для 3, 4, ..., $M-1$ слова. Некоторые слова выбрасываются по несколько раз.



После выбора слова $\mathbf{c}_{M-1}$ выброшено уже $(M-1)\sum\limits_{i=1}^{d-1}\binom{n}{i}$ слов, находящихся на расстоянии менее чем d от $\mathbf{c}_1, \ldots, \mathbf{c}_{M-1}$.

Граница Гилберта

Если

$$2^n - (M-1)\sum_{i=1}^{d-1}\binom{n}{i} > 0$$

то это значит, что ''зазор'' есть! То есть можно выбрать еще одно слово $\mathbf{c}_M$, такое что $d(\mathbf{c}_i, \mathbf{c}_M) \geq d$ для $i = 1..n-1$. То есть код (n, M, d) построен!

Теорема

Если

$$R < 1 - \frac{1}{n}\log_2\sum_{i=1}^{d-1}\binom{n}{i},$$

то код с параметрами (n, M, d) *существует.*