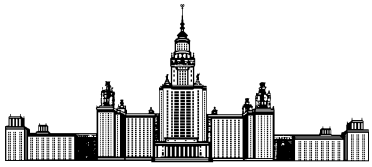
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики
Кафедра Математических Методов Прогнозирования

Дипломная работа

«Математические модели дезинформации»

Выполнил:
студент 5 курса 517 группы
Куракин Александр Владимирович

Научный руководитель:
д.ф-м.н., профессор
Леонтьев Владимир Константинович

Москва, 2014

Содержание

1. Введение

Рассмотрим понятие дезинформации. В содержательном смысле под дезинформацией понимается искажение данных, сознательное или случайное. Также под дезинформацией можно понимать некое несоотвествие при восприятии информации субъектом (сообщающим информацию) и объектом (принимающим информацию).

Таким образом параметры такой ситуации — следующие:

1) Набор исходных данных, описывающих «состояние» той или иной предметной области. Это может быть метеорологическая, медицинская, военная и другая ценная информация. Для того, чтобы информацию можно было понять, передать, воспроизвести, сравнить, ее представляют (записывают) с помощью определенного языка. Когда речь идет об общении, то это, например, русский язык. Машины хранят информацию и «общаются» на языке «ноликов и единичек» — там все данные записываются с помощью этих двух знаков. Важно, что при искажении представления (записи) информации ее смысл значимо меняется.

2) Способы искажения этих данных. Например, при разговоре по телефонной линии могут возникнуть помехи, и тогда абоненты не услышат речь собеседника. Причиной случайной дезинформации (искажения) обычно являются помехи связи, которые описываются определенными законами. Если же речь идет о сознательной дезинформации, то она должна иметь вполне легитимный характер, чтобы дезинформация осталась незаметной.

3) Набор конечных данных, описывающих результат искажения исходных данных. Это то, что мы видим, слышим в конечном итоге.

Часто мы не знаем, искажены ли исходные данные. Классический пример — фраза «казнить нельзя помиловать», в которой допустима запятая в любом месте, но смыслы фразы при этом противоположны.

Соотношение между информацией и дезинформацией в чисто формальных рамках абсолютно не определено. В то же время, если $F(x)$ — произвольная булева функция, а $N_F = \{\, x\colon\ F(x) = 1 \,\}$ — множество ее единиц, то можно считать, что если при передаче слова $x \in N_F$ получатель получает слово $y \notin N_F$, то «произошла ошибка дезинформации».

Рассмотрим процесс передачи информации от источника к получателю [2, параграф 1.1]:

1. Источник передает информацию кодеру.
2. Кодер преобразует информацию (кодирует) в вид, передаваемый по каналу.
3. Информация передается по каналу (и, возможно, искажается) и поступает в декодер.

4. Декодер восстанавливает (если это возможно) информацию. На этом шаге необходимо получить информацию в том виде, в котором она отправлялась по каналу.

5. Информация преобразуется в вид, понятный получателю.

Обсудим эту схему подробнее.

Некоторое конечное множество будем называть *алфавитом*. Тогда исходная информация — *слова* в данном алфавите. В вычислительной технике и связи алфавитом обычно является множество бит $B = \{ 0, 1 \}$. Тогда под словами понимаются конечные последовательности бит.

Некоторые, но не все, слова считаются *кодом*. Каждому исходному слову сопоставляется *кодовое слово*. При передаче по каналу слова искажаются, но так как не все слова считаются кодовыми, то есть корректными, то при искажении мы сможем заметить дезинформацию.

Например, опишем *двоичный код проверки на четность* [3, пример 1.1.4], [7, пример 1.3]. Алфавит — множество $B = \{ 0, 1 \}$, слова — конечные двоичные векторы. Кодовыми словами, или кодом, считаются слова с четным числом единиц. Если при передаче по каналу слово исказится так, что изменится его четность, то мы узнаем об искажении. Ясно, что код является не очень надежным, так как уже при изменении двух бит искажение распознать нельзя.

Противоположный пример — *двоичный код констант* [3, пример 1.1.3], состоящий из двух слов:

$$\mathbf{0} = (0, 0, \ldots, 0), \qquad \mathbf{1} = (1, 1, \ldots, 1).$$

С одной стороны, он надежный, так как не обнаружит искажения только в том случае, когда исказятся все биты исходного слова, что в большинстве каналов редкость. С другой, с помощью этого кода можно закодировать всего два слова.

Итак, исходному слову сопоставляется кодовое слово, и последнее передается по каналу. При передаче по каналу кодовое слово, вообще говоря, искажается. Искажение данных каналом может быть различное. Возможно выпадение букв, вставка букв, изменение букв. Если речь идет о двоичном коде, то может измениться длина слова, произойти замена бит вида $0 \mapsto 1$, $1 \mapsto 0$.

Каналы условно можно разделить на *помехи* и *сознательную дезинформацию* (например, [13]).

В первом случае речь идет о помехах связи, искажениях под физическим воздействием среды. Зачастую известны характеристики таких каналов. Например, при передаче по проводам часто теряются последовательности букв.

Во втором — о преднамеренных попытках человека исказить передаваемые данные.

Заметим, что в этом случае искажение должно выглядеть правдоподобным, чтобы получатель не заметил искажения. Например, если используется код проверки на четность, и злоумышленник об этом знает, то наверняка после искажения проверочная буква пересчитывается, и искаженные данные будут корректными.

Одним из хорошо изученных каналов является *двоичный симметричный канал* [1, параграф 1.1], [7, параграф 1.3]. При передаче по этому каналу двоичных слов, обычно они остаются неизменными, но с небольшой вероятностью биты инвертируются, причем независимо. Таким образом канал характеризуется вероятностью инвертации бита p, при этом буквы v_i исходного слова и v'_i искаженного связаны вероятностями

$$P(v'_i = 1|v_i = 1) = P(v'_i = 0|v_i = 0) = 1 - p,$$

$$P(v'_i = 1|v_i = 0) = P(v'_i = 0|v_i = 1) = p.$$

Другой канал — *канал, допускающий не более t ошибок* [3, параграф 1.1.2], [7, параграф 1.3]. При передаче по данному каналу в слове искажается (произвольным образом) не более t букв.

Декодер получает по каналу связи искаженное слово, при декодировании которого необходимо получить «эквивалентное» исходному слово. Очевидно, что в случае отсутствия искажений это процесс, обратный кодированию.

В общем случае верное декодирование невозможно. Тогда пытаются получить «максимально похожую на исходную» информацию и понять, имеет ли место дезинформация. Считается, что произошла дезинформация, если полученное слово не является кодовым, а декодирование осуществляется по ближайшему кодовому слову (то есть слову, которое отличается от данного минимальным количеством букв) [3, параграф 1.1.2].

Коды обладают различными характеристиками. Основное требуемое свойство — верное декодирование искаженных слов, то есть исправление ошибок. Обычно код выбирается в зависимости от того, как искажается информация при передаче по каналу. Для некоторых каналов возможно задать код, который обнаруживает искажения этого канала и может восстановить исходную информацию.

В данной работе рассмотрены примеры каналов, их формализация. Наряду с каналами рассмотрены и коды, используемые при передаче по этим каналам. Рассмотрены свойства кодов, в особенности, способность обнаруживать и исправлять ошибки данным кодом в данном канале.

В частности, будут рассмотрены коды, полезные при передаче по каналу, который инвертирует слова. Если слова — двоичные векторы, то *инвертацией*, или *отрицанием*, этого вектора называется вектор, полученный заменой всех 0 на 1, а всех 1 — на 0. Канал будем называть *инвертирующим*, если, в числе прочих искажений, он способен инвертиртировать передаваемое по нему слово.

Код, предложенный для кодирования при передаче по инвертирующему каналу, имеет следующее описание. Будем говорить, что двоичный код является *кодом, не допускающим инвертации*, если он не содержит отрицания кодовых слов:

$$\forall \mathbf{v} \in V \quad \overline{\mathbf{v}} \notin V.$$

Будут рассмотрены свойства данных кодов, в частности, обнаружение и исправление ими ошибок, число таких кодов, способ кодирования.

Один из разделов посвящен линейным кодам, то есть кодам, которые являются линейными пространствами над полем. Рассмотрены их свойства, преимущества. Отдельно рассматриваются линейные коды, не допускающие инвертации. Сформулирован критерий недопустимости инвертации линейным кодом. Рассматривается вопрос числа таких кодов, их мощности.

Также в данной работе рассматривается построение кодов как можно большой мощности, обладающих определенными свойствами. В частности, рассмотрены *коды, стойкие к угрозам*. В данной модели код должен обладать следующим свойством: при передаче по каналу кодовые слова не попадают в фиксированные «запретные» множества. Рассмотрен вопрос о максимальной его мощности.

2. Коды, исправляющие ошибки

2.1. Передача данных по каналу

Дадим несколько определений, формализующих процесс передачи данных. Заметим, что в классической литературе подходы к описанию процесса передачи данных похожи, поэтому в этом разделе ссылки на источники не приводятся.

Некоторое конечное множество Ω мощности $q = |\Omega|$ будем называть *алфавитом*.

Словом длины n в алфавите Ω будем считать вектор длины n:

$$\mathbf{a} = (a_1, \ldots, a_n) \in \Omega^n.$$

Компоненты $a_i \in \Omega$, $i \in \overline{1, n}$, суть *буквы*.

Множество всех слов длины $n \in \mathbb{N}$ в алфавите Ω обозначим через Ω^n. Множество всех слов в алфавите Ω обозначим через Ω^*.

Словарная функция на Ω^* — отображение множества слов Ω^* в себя.

Определение 1. *Кодом длины n в алфавите* Ω называется произвольное подмножество

$$V \subseteq \Omega^n.$$

Через $C = |V|$ обозначим *мощность кода*. При этом, элементы множества V называются *кодовыми словами*.

Например, *код проверки на четность* [3, пример 1.1.4], [7, пример 1.3] формально задается следующим образом. Рассмотрим алфавит $(\Omega, \cdot)$ с групповой операцией $\cdot$ и единицей e. Код проверки на четность длины n состоит из кодовых слов вида

$$\mathbf{v} = (v_1, \ldots, v_{n-1}, v_n), \quad v_1 \cdot \ldots \cdot v_{n-1} v_n = e,$$

то есть $v_n = (v_1 \cdot \ldots \cdot v_{n-1})^{-1}$. В двоичном случае ($\Omega = \mathrm{GF}(2)$) данное определение и означает, что код состоит из слов с четным числом единиц.

Обратимся к кодам, не допускающим инвертации. В качестве алфавита используются биты $B = \{0, 1\}$. Слова — конечные двоичные векторы. Код, предложенный для кодирования при передаче по инвертирующему каналу, имеет следующее

Определение 2. Будем говорить, что двоичный код является *кодом, не допускающим инвертации*, если он не содержит отрицания кодовых слов:

$$\forall\, \mathbf{v} \in V \quad \overline{\mathbf{v}} \notin V.$$

Например, следующий код не допускает инвертации:

$$\{(000), (001), (100), (101)\}.$$

В то же время, добавление отрицания любого кодового слова лишит код этого свойства. Например, код

$$\{(000), (001), (100), (101), (110)\},$$

не является кодом, не допускающим инвертации. Код проверки на четность длины 3

$$\{(000), (011), (101), (110)\}$$

не допускает инвертации. Однако код проверки на четность длины 4

$$\{(0000), (0011), (0101), (0110), (1001), (1010), (1100), (1111)\}$$

не является кодом, не допускающим инвертации: он содержит отрицание своих элементов, например, (1111) — отрицание (0000).

Вернемся к обсуждению схемы передачи данных.

Определение 3. *Кодированием слов Ω^* согласно коду V* называется взаимно-однозначное отображение множества слов Ω^* в код V.

Например, как закодировать двоичные слова длины n согласно коду проверки на четность? Очевидно, необходимо использовать код проверки на четность длины $n+1$. При этом кодовое слово образуется из исходного дописыванием 0, если число единиц в слове четно, и 1, если нечетно. Также см. [7, таблица 1.1].

Формализовав процесс кодирования, обратимся к процессу передачи по каналу.

Определение 4. *Каналом* называется многозначное отображение кода во множество слов:

$$\Psi(\mathbf{v}) = \{\ \psi_1(\mathbf{v}), \ldots, \psi_m(\mathbf{v})\ \}, \qquad \mathbf{v} \in V, \quad \psi_k(\mathbf{v}) \in \Omega^*, \quad k = \overline{1, m}.$$

Последнее означает следующее: при передаче по каналу кодовое слово $\mathbf{v}$ переходит в одно из слов $\psi_i(\mathbf{a})$, $i \in \overline{1, m}$.

Рассмотрим примеры каналов. Пусть используется двоичный алфавит $\Omega = B = \{\ 0, 1\ \}$. *Весом Хэмминга слова длины n* называется число его ненулевых компонент:

$$\|\mathbf{a}\| = |\{\ i \in \overline{1, n}\colon\ a_i \neq 0\ \}|, \qquad \mathbf{a} = (a_1, \ldots, a_n) \in \Omega^n.$$

Шаром радиуса t с центром в нуле назовем множество слов веса не более t:

$$S_t(\mathbf{0}) = \{\ \mathbf{a} \in \Omega^n\colon\ \ \|\mathbf{a}\| \leq t\ \}.$$

1) *Канал, допускающий не более t ошибок* в двоичном случае описывается как канал, в котором происходит искажение бит (букв) следующего вида:

$$0 \mapsto 1, \qquad 1 \mapsto 0,$$

причем происходит не более t искажений, может быть описан в виде следующих словарных функций:

$$\psi_{\mathbf{a}}(\mathbf{v}) = \mathbf{v} \oplus \mathbf{a}, \qquad \mathbf{a} \in S_t(\mathbf{0}).$$

Таким образом классический канал может быть задан с помощью m булевых функций ($m = |S_t(\mathbf{0})| = \sum_{i=0}^{t} \binom{n}{i}$), каждая из которых представляет собой сложение по модулю 2 исходного слова и ошибки. В этом случае говорят об *аддитивном канале* (например, [6]).

2) Рассмотрим *канал с разделением ошибок*. Он искажает двоичные слова согласно классической схеме

$$0 \mapsto 1, \quad 1 \mapsto 0,$$

при этом в определенном интервале букв ошибки исключены. Векторы ошибок $\mathbf{a_k} = (a_{k1}, \ldots, a_{kn})$, $k = \overline{1,m}$, являются решениями системы

$$\begin{cases} a_{k1} + a_{k2} + \cdots + a_{kk} & \le 1 \\ a_{k2} + a_{k3} + \cdots + a_{k,k+1} & \le 1 \\ \cdots & \\ a_{k,n-k+1} + a_{k,n-k+2} + \cdots + a_{k,n} & \le 1 \end{cases}.$$

Таким образом канал есть

$$\Psi(\mathbf{v}) = \{\psi_1(\mathbf{v}), \ldots, \psi_m(\mathbf{v})\}, \qquad \psi_k(\mathbf{v}) = \mathbf{v} \oplus \mathbf{a_k}, \quad k \in \overline{1,m}.$$

Рассмотрим пример. Если $n = 4$, $k = 2$, то описанные неравенства имеют вид:

$$\begin{cases} a_{k1} + a_{k2} \le 1 \\ a_{k2} + a_{k3} \le 1 \\ a_{k3} + a_{k4} \le 1 \end{cases}.$$

Множество решений образуют код

$$\{(0000), (1000), (0100), (0100), (0010), (1010), (0101), (1001)\}.$$

3) Рассмотрим *канал с транспозицией не более одной пары соседних букв*, то есть преобразования вида

$$\psi_i(v_1 \ldots v_{i-1} v_i v_{i+1} v_{i+2} \ldots v_n) = v_1 \ldots v_{i-1} v_{i+1} v_i v_{i+2} \ldots v_n, \qquad i \in \overline{1, n-1}.$$

Формально данные преобразования задаются с помощью матриц, получаемых из единичной перестановкой двух строк:

$$A_1 = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}, \; A_2 = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}, \; \dots, \; A_{n-1} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & \dots & 1 & 0 \end{pmatrix}.$$

Тогда, как нетрудно проверить, $\psi_i(\mathbf{v}) = \mathbf{v}A_i$, а канал задается отображением

$$\Psi(\mathbf{v}) = \{\, \mathbf{v}A_1, \dots, \mathbf{v}A_{n-1} \,\}.$$

4) Канал, который либо оставляет слово неизменным, либо инвертирует его, задается отображением

$$\Psi(\mathbf{v}) = \{\psi_1(\mathbf{v}), \psi_2(\mathbf{v})\}, \qquad \psi_1(\mathbf{v}) = \mathbf{v}, \quad \psi_2(\mathbf{v}) = \overline{\mathbf{v}}.$$

5) Зададим канал, допускающий не более t ошибок ($t < n$) и, возможно, инвертирующий слово. Наряду с функциями

$$\psi_{\mathbf{a}}(\mathbf{v}) = \mathbf{v} \oplus \mathbf{a}, \qquad \mathbf{a} \in S_t(\mathbf{0}),$$

имеем инвертирующую функцию

$$\psi_{\mathbf{1}}(\mathbf{v}) = \overline{\mathbf{v}} = \mathbf{v} \oplus \mathbf{1},$$

где $\mathbf{1} = (1, 1, \dots, 1) \in B^n$. Канал может быть задан с помощью $(m+1)$-й булевой функции аналогично стандартному случаю.

6) Рассмотрим алфавит $B' = B \cup \Lambda = \{\, 0, 1, \Lambda \,\}$, то есть добавим букву «значение не определено». Рассмотрим новый канал: пусть при передаче по этому каналу происходит либо искажение не более одного бита

$$0 \mapsto 1, \quad 1 \mapsto 0,$$

либо не более одного «стирания»

$$0 \mapsto \Lambda, \quad 1 \mapsto \Lambda.$$

Зададим этот канал формально. Действительно, пусть

$$\varphi_i(\mathbf{v}) = \mathbf{v} \oplus \mathbf{e_i}, \qquad (\mathbf{e_i})_j = \begin{cases} 1, & \text{если } j = i \\ 0, & \text{иначе} \end{cases}, \qquad i = \overline{1, n}.$$

и

$$\psi_i(v_1 \dots v_{i-1} v_i v_{i+1} \dots v_n) = v_1 \dots v_{i-1} \Lambda v_{i+1} \dots v_n, \qquad i = \overline{1,n}.$$

Тогда многозначное отображение

$$\Psi(\mathbf{v}) = \{\, \varphi_0(\mathbf{v}), \dots, \varphi_n(\mathbf{v}), \psi_0(\mathbf{v}), \dots, \psi_n(\mathbf{v}) \,\}$$

представляет собой описание этого канала.

Более подробно классификация каналов описывается в [8, глава 4].

После получение информации по каналу необходимо восстановить исходную информацию. Декодирование — восстановление исходного кодового слова по искаженному. Процесс декодирования можно осуществить по *таблице декодирования*:

$\mathbf{v_1}$	$\mathbf{v_2}$	$\cdots$	$\mathbf{v_C}$
$\psi_1(\mathbf{v_1})$	$\psi_1(\mathbf{v_2})$	$\cdots$	$\psi_1(\mathbf{v_C})$
$\psi_2(\mathbf{v_1})$	$\psi_2(\mathbf{v_2})$	$\cdots$	$\psi_2(\mathbf{v_C})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\psi_m(\mathbf{v_1})$	$\psi_m(\mathbf{v_2})$	$\cdots$	$\psi_m(\mathbf{v_C})$

(в этой таблице в i-м столбце перечисляются все возможные искажения слова $\mathbf{v_i}$). Если полученное по каналу слово встречается в l-м столбце, то вектор $\mathbf{v_l}$ следует считать исходным. Однако таких столбцов может быть несколько, тогда декодирование неоднозначно.

Теперь можем уточнить схему передачи данных:

1. Кодер получает информацию, то есть слово $\mathbf{a} \in \Omega^*$.
2. Кодер кодирует слово, получая кодовое слово $\mathbf{v} \in V$. Заметим, что код $V \subseteq \Omega^n$ известен как кодеру, так и декодеру.
3. Кодовое слово передается по каналу и, возможно, искажается отображением Ψ. Декодер получает некое слово $\mathbf{v}' = \psi_k(\mathbf{v}) \in \Omega^*$, $k \in \overline{1,m}$, возможно, не равное исходному ($\mathbf{v}' \neq \mathbf{v}$) (и, возможно, не являющееся кодовым ($\mathbf{v}' \notin V$)).
4. Декодер восстанавливает информацию, то есть получает слово $\mathbf{v_0} \in V$. Если кодовое слово восстановлено верно, то $\mathbf{v_0} = \mathbf{v}$.
5. Декодер преобразует полученную информацию в вид, понятный получателю и «эквивалентный» исходной информации. Если на предыдущем шаге кодовое слово было $\mathbf{v}$ восстановлено верно, то на данном шаге получаем исходное слово $\mathbf{a}$.

2.2. Параметры кода

В этом параграфе (и до конца главы) описываются классические характеристики кода. Характеристики кода рассматриваются во многих источниках по теории кодирования. Кратко они описываются, например, в [3, глава 1].

Пусть имеется алфавит Ω мощности $q = |\Omega|$ и код V длины n в этом алфавите.

Следующая характеристика аналогична мощности кода, но учитывает и мощность алфавита.

Определение 5. *Комбинаторная размерность кода* суть следующая величина:

$$k = \log_{|\Omega|} |V|.$$

(Комбинаторная размерность — число не обязательно рациональное.)

Введем «меру различия» кодовых слов.

Определение 6. *Расстоянием Хэмминга* между словами одной длины называется число компонент, в которых они отличаются:

$$d(\mathbf{a}, \mathbf{b}) = |\{\, i \in \overline{1,n} \colon\ a_i \neq b_i \,\}|, \qquad \mathbf{a} = (a_1, \ldots, a_n), \mathbf{b} = (b_1, \ldots, b_n) \in \Omega^n.$$

Размерность кода характеризует долю информации, передаваемую кодовым словом длины n, поэтому величину k/n называют *скоростью кода* [11]. Если $k/n \approx 1$ то говорят, что код с высокой скоростью передачи информации, если $k/n \ll 1$ — что с низкой.

Спектр расстояний между словами кода играет важную роль. Чем меньше расстояние, тем меньше изменений букв требуется для искажения, превращающего одно кодовое слово в другое (случай, когда мы не можем распознать дезинформацию). Отсюда следующее

Определение 7. *Кодовым расстоянием кода* (мощности не менее 2) называется минимальное расстояние между его кодовыми словами:

$$d = d(V) = \min \{\, d(\mathbf{u}, \mathbf{v}) \colon\ \mathbf{u}, \mathbf{v} \in V, \quad \mathbf{u} \neq \mathbf{v} \,\}.$$

Для кодов с известными значениями указанных параметров приняты следующие обозначения: $(n, C)_\Omega$-*код*, $(n, C)_q$-*код*, $[n, k]_\Omega$-*код*, $[n, k]_q$-*код*, $[n, k, d]_\Omega$-*код*, $[n, k, d]_q$-*код*.

Например, двоичный код проверки на четность является $[n, n-1, 2]_\Omega$-кодом. Действительно, если $\mathbf{u}, \mathbf{v} \in \Omega^n$ и $d(\mathbf{u}, \mathbf{v}) = 1$, то одновременное вхождение этих векторов в код невозможно, то есть $d \geq 2$. С другой стороны, код проверки на четность содержит слова $\mathbf{u} = (e, \ldots, e, a, a^{-1})$ и $\mathbf{v} = (e, \ldots, e, a, e, a^{-1})$, где $a \in \Omega \setminus e$, и $d \leq d(\mathbf{u}, \mathbf{v}) = 2$. Отсюда кодовое расстояние $d = 2$.

Код, не допускающий инвертации, имеет ограничение $d \leq n - 1$. Действительно, если $d = n$, то в коде длины n присутствуют $\mathbf{u}$ и $\mathbf{v}$ такие, что $d(\mathbf{u}, \mathbf{v}) = n$. Тогда $\mathbf{u} = \overline{\mathbf{v}}$ и $\mathbf{v}, \overline{\mathbf{v}} \in V$. Мощность кода, не допускающего инвертации, не превосходит, тем самым, половины максимальной мощности произвольного кода.

2.3. Способность кода обнаруживать и исправлять ошибки

Важное свойство кодов — способность обнаруживать и исправлять ошибки. Сформулируем соответствующие определения.

Определение 8. *Код V обнаруживает ошибки данного канала*, если любое кодовое слово $\mathbf{v} \in V$ в случае искажения каналом перестает быть кодовым, то есть $\Psi(\mathbf{v}) \notin V$.

Например, код проверки на четность обнаруживает ошибки канала, изменяющего сумму букв кодового слова. Коды, не допускающие инвертации, обнаруживают ошибки канала, который либо оставляет слово неизменным, либо инвертирует его.

Определение 9. *Код V исправляет ошибки данного канала*, если по значению искаженному значению $\Psi(\mathbf{v})$ можно восстановить $\mathbf{v}$ для любого кодового слова $\mathbf{v} \in V$.

Например, код проверки на четность исправляет ошибки канала, изменяещего букву с фиксированным номером. Код, не допускающий инвертации, исправляет ошибки канала, который либо оставляет слово неизменным, либо инвертирует его.

Для данного кода $V = \{\mathbf{v_1}, \ldots, \mathbf{v_C}\}$ и канала $\Psi(\mathbf{v}) = \{\psi_1(\mathbf{v}), \ldots, \psi_m(\mathbf{v})\}$ построим таблицу декодирования:

$\mathbf{v_1}$	$\mathbf{v_2}$	$\cdots$	$\mathbf{v_C}$
$\psi_1(\mathbf{v_1})$	$\psi_1(\mathbf{v_2})$	$\cdots$	$\psi_1(\mathbf{v_C})$
$\psi_2(\mathbf{v_1})$	$\psi_2(\mathbf{v_2})$	$\cdots$	$\psi_2(\mathbf{v_C})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\psi_m(\mathbf{v_1})$	$\psi_m(\mathbf{v_2})$	$\cdots$	$\psi_m(\mathbf{v_C})$

Отметим теперь, что если столбцы матрицы декодирования не пересекаются, то код V исправляет ошибки Ψ, так как исходное кодовое слово определяется столбцом (его номером) матрицы декодирования. Итак, матрица декодирования показывает, исправляет ли код ошибки канала.

Канал $\Psi(\mathbf{v})$ как множество функций $\{\psi_1(\mathbf{v}), \ldots, \psi_m(\mathbf{v})\}$ задает на V отношение

$$\mathbf{u} \sim \mathbf{v} \quad \Leftrightarrow \quad \exists k\colon\ \psi_k(\mathbf{u}) = \psi_k(\mathbf{v}), \qquad \mathbf{u}, \mathbf{v} \in V.$$

Оно является отношением толерантности, то есть рефлексивно и симметрично. Соответственно, каждое кодовое слово $\mathbf{v} \in V$ порождает класс толерантности $R_{\mathbf{v}} \subseteq V$.

Рассмотрим пример. Пусть канал задается отображением

$$\Psi(\mathbf{v}) = \{\psi_1(\mathbf{v}), \psi_2(\mathbf{v})\}.$$

Тогда каждое кодовое слово $\mathbf{v} \in V$ порождает класс толерантности:

$$R_{\mathbf{v}} = \{\ \mathbf{u} \in V\colon\ \psi_1(\mathbf{u}) = \psi_1(\mathbf{v}) \vee \psi_1(\mathbf{u}) = \psi_2(\mathbf{v}) \vee \psi_2(\mathbf{u}) = \psi_1(\mathbf{v}) \vee \psi_2(\mathbf{u}) = \psi_2(\mathbf{v})\ \}.$$

Таким образом $R_{\mathbf{v}}$ задается решением (относительно **u**) следующей совокупности уравнений:

$$\left[\begin{array}{l} \psi_1(\mathbf{u}) = \psi_1(\mathbf{v}) \\ \psi_1(\mathbf{u}) = \psi_2(\mathbf{v}) \\ \psi_2(\mathbf{u}) = \psi_1(\mathbf{v}) \\ \psi_2(\mathbf{u}) = \psi_2(\mathbf{v}) \end{array}\right.$$

В терминах данного отношения критерий исправления кодом ошибок канала формулируется в виде следующего утверждения.

Утверждение 1. Код исправляет ошибки канала тогда и только тогда, когда каждый из описанных классов толерантности состоит из единственного элемента.

Код исправляет ошибки описанного канала тогда и только тогда, когда совокупность имеет единственное решение, для каждого $\mathbf{v} \in V$.

2.3.1. Способность кода обнаруживать и исправлять t ошибок

Рассмотрим канал, искажающий не более t букв кодового слова. Тогда обнаружение ошибок и их исправление формализуется следующим образом.

Шаром радиуса t *с центром в слове* **a** называется множество

$$S_t(\mathbf{a}) = \{\, \mathbf{b} \in \Omega^n :\ d(\mathbf{a}, \mathbf{b}) \le t \,\}.$$

Шаром радиуса t описывается множество слов, которые можно получить, изменив в слове-центре не более t его букв.

Определение 10. Говорят, что *код* V *обнаруживает* t *ошибок*, если шары радиуса t с центрами в кодовых словах содержат единственное кодовое слово:

$$\forall\, \mathbf{v} \in V \quad S_t(\mathbf{v}) \cap V = \{\mathbf{v}\}.$$

Определение 11. Говорят, что *код* V *исправляет* t *ошибок*, если шары радиуса t с центрами в кодовых словах взаимно не пересекаются:

$$\forall\, \mathbf{u}, \mathbf{v} \in V :\ \mathbf{u} \ne \mathbf{v} \quad S_t(\mathbf{u}) \cap S_t(\mathbf{v}) = \varnothing.$$

Например, код $V = \{(000000), (111000), (000111)\}$ обнаруживает две ошибки и исправляет одну ошибку. Действительно, инвертируя любые два бита любого из кодовых слов, мы не получим другое кодовое слово, то есть канал не может исказить информацию незаметно для получателя. Однако изменением трех бит можно, например, получить (000000) из (111000). Если изменить любой бит любого кодового слова, то мы получим

искаженное слово, которое невозможно получить искажением одного бита другого кодового слова. Тем самым, мы всегда можем восстановить исходное кодовое слово (таковым будет ближайшее к искаженному).

Способность кода обнаруживать и исправлять ошибки напрямую связаны с его кодовым расстоянием [3 (параграф 1.1.2)].

Утверждение 2. Код V обнаруживает t ошибок если и только если $d(V) > t$.

Утверждение 3. Код V исправляет t ошибок если и только если $d(V) > 2t$.

Пример согласуется с этими утверждениями, в нем $d(V) = 3$.

Известно, что параметры кода не могут быть произвольными. Условия, которыми связаны параметры кода, называются *границами кода*.

Важной является следующая граница:

Утверждение 4 (граница Синглтона)**.** В произвольном $[n, k, d]_\Omega$-коде выполняется неравенство:

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

Выделяется особый класс кодов, где эта граница достигается: произвольный $[n, k, n-k+1]_\Omega$-код называется *МДР-кодом*. МДР коды описываются, например, в [1, глава 11].

Позже будет рассмотрен вопрос об МДР-кодах, не допускающих инвертации.

3. Линейные коды

Опишем широко используемый класс кодов — *линейные коды*. Более подробно можно прочесть в [2, глава 3], [1, глава 1]. Основные результаты содержатся в [3, глава 2], [10].

Определение 12. Код M, который является линейным пространством над полем P, называется *линейным кодом над* P.

Иными словами, кодовые слова линейного кода можно складывать, умножать на элементы из P, а линейные комбинации кодовых слов также будут кодовыми словами. Заметим, что в линейном коде присутствует нулевой вектор $\mathbf{0}$.

Далее линейным двоичным кодом будем называть линейный над $\mathrm{GF}(2)$ код в алфавите $\mathrm{GF}(2)$. Например, следующий код является линейным двоичным кодом:

$$\{(00000), (01011), (10110), (11101)\}.$$

Вот некоторые преимущества линейных кодов [10, параграф 2.1]:

1. Нахождение расстояний в коде, как и кодового расстояния, проще общего случая.
2. Кодирование проще и требует меньше памяти.
3. Проще определить, обнаруживаются, исправляются ли кодом ошибки данного канала.
4. Вероятность верного декодирования вычисляется проще.
5. Существуют эффективные методы декодирования.

Некоторые из этих пунктов рассматриваются далее.

Обратимся к кодам, не допускающим инвертации. Есть ли среди них линейные? Да, например, приведенный выше код не допускает инвертации. Позже будет сформулирован соответствующий критерий.

3.1. Расстояния в линейном коде

Напомним, что *весом Хэмминга слова длины* n называется число его ненулевых компонент:

$$\|\mathbf{a}\| = |\{\, i \in \overline{1,n} \colon\ a_i \neq 0 \,\}|, \qquad \mathbf{a} = (a_1, \ldots, a_n) \in \Omega^n.$$

Для линейного кода справедливы следующие полезные утверждения [3, глава 2]:

Утверждение 5. Расстояние между словами линейного кода суть вес их разности:

$$\forall\, \mathbf{u}, \mathbf{v} \in M \quad d(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|\,.$$

Таким образом для линейного кода справедливо отношение:

$$\forall\, \mathbf{v} \in M \quad \|\mathbf{v}\| = d(\mathbf{v}, \mathbf{0}),$$

а для нахождения кодового расстояния не надо перебирать все пары кодовых слов.

Утверждение 6. Кодовое расстояние линейного кода определяется минимальным весом среди его ненулевых слов:

$$d(M) = \min\{\, \|\mathbf{v}\| :\ \mathbf{v} \in M \setminus \mathbf{0}\,\}.$$

Например, в приведенном выше линейном коде

$$\{(00000), (01011), (10110), (11101)\}.$$

имеем $d = 3$. Из этого следует, согласно уже известным утверждениям, что код обнаруживает две ошибки и исправляет одну ошибку.

Утверждение 7. Линейный $[n, k]_P$-код M над полем P есть подпространство размерности k. Мощность этого подпространства равна 2^k.

В частности, данный код является $[5, 2, 3]_B$-кодом и, тем самым, является подпространством размерности 2. Оно задается, например, базисом $((01011), (10110))$.

3.2. Задание линейного кода с помощью матриц

3.2.1. Проверочная матрица линейного кода

Следующее утверждение позволяет описать линейный код над полем с помощью матрицы над ним (доказательства следует искать в [3, глава 2]).

Утверждение 8. Линейный $[n, k]_P$-код над полем P есть ядро некоторой матрицы над P:

$$\exists H \in P^{l\times n}:\ M = \{\, \mathbf{v} \in P^n :\ H\mathbf{v} = \mathbf{0}\,\}.$$

При этом rank $H = n - k$. Данную матрицу будем называть *проверочной матрицей кода* M.

По проверочной матрице можно найти кодовое расстояние линейного кода.

Утверждение 9. Пусть H — проверочная матрица линейного кода M, и $\varkappa(H)$ — наибольшее натуральное число такое, что любая система из $\varkappa(H)$ столбцов матрицы H является линейно независимой. Тогда

$$d(M) = \varkappa(H) + 1.$$

Любую $r \times n$-матрицу ранга r над полем P можно элементарными преобразованиями строк и перестановкой столбцов привести к виду

$$H_0 = \begin{pmatrix} r_{11} & \dots & r_{1k} & -e & 0 & \dots & 0 \\ r_{21} & \dots & r_{2k} & 0 & -e & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ r_{l1} & \dots & r_{lk} & 0 & 0 & \dots & -e \end{pmatrix} = (\ \overline{H}, -E_{l\times l}\ )_{l\times n}.$$

Если речь идет о проверочной матрице кода, то матрицы такого вида будем называть *проверочными матрицами в стандартном виде*. Заметим, что линейные коды совпадают тогда и только тогда, когда совпадают их проверочные матрицы в стандартном виде.

Обладая аппаратом проверочных матриц, можем сформулировать критерий того, что линейный код не допускает инвертации.

Утверждение 10. Пусть M — линейный двоичный код, и H — его проверочная матрица, приводимая к стандартному виду $H_0 = (\ \overline{H}, -E_{l\times l}\ )$. Тогда следующие утверждения эквивалентны:

1. Код M не допускает инвертации,
2. Вектор $\mathbf{1} = (1, 1, \dots, 1)$ не входит в M,
3. В матрице H присутствует строка с нечетным весом.
4. В матрице $\overline{H}$ присутствует строка с четным весом.

Доказательство. Докажем эквивалентность $1 \Leftrightarrow 2$. По определению код M не допускает инвертации, если и только если верна импликация:

$$\mathbf{v} \in M \quad \Rightarrow \quad \overline{\mathbf{v}} \notin M.$$

Учитывая, что H — проверочная матрица M, то имеем эквивалентное условие:

$$H\mathbf{v} = 0 \quad \Rightarrow \quad H\overline{\mathbf{v}} = 0.$$

Так как $H\overline{\mathbf{v}} = H\mathbf{v} \oplus H \cdot \mathbf{1}$, можем сложить это равенство с посылкой импликации, получим:

$$H\mathbf{v} = 0 \quad \Rightarrow \quad H \cdot \mathbf{1} = 0.$$

Снова используя аппарат проверочной матрицы, имеем

$$\mathbf{v} \in M \quad \Rightarrow \quad \mathbf{1} \in M.$$

Эквивалентность $2 \Leftrightarrow 3$ верна согласно определению произведения матрицы и вектор-столбца.

Справедливость $3 \Leftrightarrow 4$ очевидна. $\square$

3.2.2. Порождающая матрица линейного кода

Аналогичный способ построения линейного кода — с помощью его порождающей матрицы.

Определение 13. Матрица $G \in P^{m \times n}$ называется *порождающей матрицей линейного кода* M, если система ее строк порождает M.

Очевидно, что при этом справедливы соотношения $m \geq \operatorname{rank} G = \dim M = k$, и матрицу G можно выбрать так, что $m = k$.

Легко доказывается

Утверждение 11. Пусть M — линейный код длины n над полем P и $G_{m \times n}$, $H_{l \times n}$ — матрицы над полем P. Тогда следующие утверждения эквивалентны:

1. G — порождающая, а H — проверочная матрицы кода M,
2. G — порождающая матрица кода M и верны равенства
$$HG^T = \mathbf{0}, \qquad \operatorname{rank} H + \operatorname{rank} G = n.$$
3. H — проверочная матрицы кода M и верны равенства
$$HG^T = \mathbf{0}, \qquad \operatorname{rank} H + \operatorname{rank} G = n.$$

Заметим, что если код имеет проверочную матрицу H в стандартном виде, то данным условиям удовлетворяет матрица

$$G_0 = (\, E_{k \times k},\, \overline{H}^T)_{k \times n},$$

которая называется *порождающей матрицей кода M в стандартном виде*.

Обратимся к линейным двоичным кодам. Согласно сформулированному раннее критерию, линейный двоичный код M не допускает инвертации тогда и только тогда, когда в «основной» части $\overline{H}$ проверочной матрицы стандартного вида присутствует строка с четным весом. Тогда, как видно из определения, в «основной» части $\overline{H}^T$ порождающей матрицы коды M в стандартном виде имеет столбец с четным весом.

Линейные коды совпадают тогда и только тогда, когда совпадают их порождающие матрицы в стандартном виде. Учитывая аналогичное свойство проверочных матриц, можем сформулировать более широкое

Утверждение 12. Следующие утверждения эквивалентны:

1. Линейные коды совпадают.
2. Совпадают их проверочные матрицы в стандартном виде.
3. Совпадают их порождающие матрицы в стандартном виде.

3.2.3. Линейные двоичные МДР-коды

Напомним, что $[n, k, n-k+1]_\Omega$-код называется МДР-кодом. Зададимся вопросом: какие из кодов, не допускающих инвертации, являются МДР кодами? Для начала рассмотрим, какие двоичные коды являются МДР-кодами. Следующее утверждение часто приводится в литературе (например, в [9]), однако его доказательство, представляющее определенный интерес, автору не встретилось и было проведено самостоятельно.

Утверждение 13. Следующие коды являются линейными двоичными МДР-кодами длины n ненулевой размерности, и только они:

1. код констант: $\{\ \mathbf{0}, \mathbf{1}\ \}$,
2. код проверки на четность: $\{\ \mathbf{v} \in \mathrm{GF}(2)^n : \ \sum_{i=1}^{n} v_i = 0\ \}$,
3. все пространство $\mathrm{GF}(2)^n$.

Доказательство. Докажем утверждение, рассматривая размерность k.

При $k = 1$ должно выполняться равенство $d = n - k + 1 = n$, линейные двоичные МДР-коды ограничиваются кодом констант.

При $k = n$ порождающая матрица кода невырождена, соответствующее матричное уравнение порождает все пространство.

Пусть теперь $1 < k < n$. Тогда канонический вид порождающей матрицы имеет вид

$$G_0 = (\ E_{k\times k},\ \overline{H}^T)_{k\times n},$$

причем в «основной» части $\overline{H}^T$ присутствует хотя бы один столбец.

Выберем $k-1$ столбец из единичной матрицы (в выбранные столбцы не войдет i-й столбец), и первый столбец из «основной» части. Система из k столбцов должна быть линейно независимой (по критерию МДР-кодов, [1, с. 310]), отсюда i-й элемент в этом столбце равен 1. Учитывая произвольность выбора i заключаем, что первый столбец «основной» части матрицы целиком состоит из единиц.

Аналогично заключаем, что второй, третий и так далее столбцы состоят целиком из единиц. Вся «основная» часть матрицы состоит из единиц.

А теперь заметим, что если взять $k-2$ столбца из единичной части и два столбца из «основной» части, то получим линейно зависимую систему, так как в любом случае два столбца из «основной» части равны. Противоречие. Значит, в «основной» части матрицы ровно один столбец (более того, целиком состоящий из единиц). Нетрудно проверить, что эта матрица является проверочной для кода проверки на четность.

Рассмотрены все возможные k и, тем самым, все линейные двоичные МДР-коды. $\square$

Наложим теперь ограничение: пусть код не допускает инвертации. Напомним, это выполняется, если и только если код не содержит вектора $\mathbf{1} = (1, 1, \ldots, 1)$. Таким образом код констант $\{\ \mathbf{0}, \mathbf{1}\ \}$ и все пространство $\mathrm{GF}(2)^n$ не подходят. Код проверки на четность длины n содержит вектор из единиц при четном n. Окончательно, линейные двоичные коды длины n, не допускающие инвертации, — в точности код, состоящий из нуля; код проверки на четность, если n нечетное.

3.3. О числе линейных двоичных кодов

Заметим, что линейность q-ичного кода эквивалентна тому, что код групповой, то есть замкнут относительно сложения по модулю q:

$$\forall\, \mathbf{a}, \mathbf{b} \in M \quad \mathbf{a} \oplus \mathbf{b} \in M.$$

Следующие утверждения описывают число линейных q-ичных кодов. Их доказательства можно найти в [14, параграф 4 главы III].

Утверждение 14. Число линейных q-ичных кодов длины n размерности k есть

$$\begin{bmatrix} n \\ k \end{bmatrix}_q = \prod_{i=1}^{k} \frac{q^{n-k+i} - 1}{q^i - 1} = \frac{(q^n - 1)(q^{n-1} - 1) \cdot \ldots \cdot (q^{n-k+1} - 1)}{(q^k - 1)(q^{k-1} - 1) \cdot \ldots \cdot (q - 1)}.$$

Утверждение 15. Число линейных q-ичных кодов длины n есть $\sum_{k=1}^{n} \begin{bmatrix} n \\ k \end{bmatrix}_q$.

Обратимся к линейным кодам, не допускающим инвертации. Исследуем вопрос: а сколько кодов, не допускающих инвертации, среди линейных двоичных кодов?

Обозначим число линейных кодов длины n размерности k, не допускающих инвертации, через $L(n, k)$.

Утверждение 16. Справедливо соотношение:

$$L(n, k) = \begin{bmatrix} n \\ k \end{bmatrix}_2 - \begin{bmatrix} n-1 \\ k-1 \end{bmatrix}_2 = 2^k \begin{bmatrix} n-1 \\ k \end{bmatrix}_2.$$

Доказательство. Число линейных двоичных кодов длины n размерности k есть $\begin{bmatrix} n \\ k \end{bmatrix}_2$. Среди них коды, содержащие вектор $\mathbf{1}$, и только они, не являются кодами, не допускающими

инвертации. Согласно [Сачков, страницы 125-126] их число есть $\begin{bmatrix} n-1 \\ k-1 \end{bmatrix}_2$, откуда следует первое равенство. Там же доказывается второе равенство. □

Отсюда следует:

$$L(n, n-1) = 2^{n-1},$$

$$L(n, n-2) = 2^{n-2}(2^{n-1}-1),$$

$$L(n, n-3) = \frac{2^{n-3}(2^{n-2}-1)(2^{n-1}-1)}{3},$$

и так далее.

Утверждение 17. Доля линейных кодов длины n размерности k, не допускающих инвертации, среди двоичных линейных кодов есть

$$\frac{2^n - 2^k}{2^n - 1}.$$

Доказательство. Искомая величина есть

$$\frac{L(n,k)}{\begin{bmatrix} n \\ k \end{bmatrix}_2} = \frac{\begin{bmatrix} n \\ k \end{bmatrix}_2 - \begin{bmatrix} n-1 \\ k-1 \end{bmatrix}_2}{\begin{bmatrix} n \\ k \end{bmatrix}_2} = 1 - \frac{\begin{bmatrix} n-1 \\ k-1 \end{bmatrix}_2}{\begin{bmatrix} n \\ k \end{bmatrix}_2} = 1 - \frac{\frac{(2^{n-1}-1)(2^{n-2}-1)\cdot\ldots\cdot(2^{n-k+1}-1)}{(2^{k-1}-1)(2^{k-2}-1)\cdot\ldots\cdot(2-1)}}{\frac{(2^n-1)(2^{n-1}-1)\cdot\ldots\cdot(2^{n-k+1}-1)}{(2^k-1)(2^{k-1}-1)\cdot\ldots\cdot(2-1)}} =$$

$$= 1 - \frac{(2^{n-1}-1)(2^{n-2}-1)\cdot\ldots\cdot(2^{n-k+1}-1)\cdot(2^k-1)(2^{k-1}-1)\cdot\ldots\cdot(2-1)}{(2^{k-1}-1)(2^{k-2}-1)\cdot\ldots\cdot(2-1)\cdot(2^n-1)(2^{n-1}-1)\cdot\ldots\cdot(2^{n-k+1}-1)} =$$

$$= 1 - \frac{2^k-1}{2^n-1} = \frac{2^n-2^k}{2^n-1}.$$

□

Проведенные рассуждения показывают, что среди линейных двоичных кодов коды, не допускающии инвертации, присутствуют, их немало. Это позволяет использовать коды с линейными свойствами, не допускающие инвертации.

Приведем таблицы, отражающие (при фиксированных n и k): число линейных кодов, число линейных кодов, не допускающих инвертации и долю линейных кодов, не допускающих инвертации, среди линейных кодов. Заметим, что эти данные подтверждены переборным алгоритмом с помощью ЭВМ.

n	k	$\begin{bmatrix} n \\ k \end{bmatrix}_2$	$L(n,k)$	Доля
1	1	1	0	0.0000
2	1	3	2	0.6667
2	2	1	0	0.0000
3	1	7	6	0.8571
3	2	7	4	0.5714
3	3	1	0	0.0000
4	1	15	14	0.9333
4	2	35	28	0.8000
4	3	15	8	0.5333
4	4	1	0	0.0000
5	1	31	30	0.9677
5	2	155	140	0.9032
5	3	155	120	0.7742
5	4	31	16	0.5161
5	5	1	0	0.0000
6	1	63	62	0.9841
6	2	651	620	0.9524
6	3	1395	1240	0.8889
6	4	651	496	0.7619
6	5	63	32	0.5079
6	6	1	0	0.0000
7	1	127	126	0.9921
7	2	2667	2604	0.9764
7	3	11811	11160	0.9449
7	4	11811	10416	0.8819
7	5	2667	2016	0.7559
7	6	127	64	0.5039
7	7	1	0	0.0000
8	1	255	254	0.9961
8	2	10795	10668	0.9882
8	3	97155	94488	0.9725
8	4	200787	188976	0.9412
8	5	97155	85344	0.8784
8	6	10795	8128	0.7529
8	7	255	128	0.5020
8	8	1	0	0.0000

n	k	$\begin{bmatrix} n \\ k \end{bmatrix}_2$	$L(n,k)$	Доля
9	1	511	510	0.9980
9	2	43435	43180	0.9941
9	3	788035	777240	0.9863
9	4	3309747	3212592	0.9706
9	5	3309747	3108960	0.9393
9	6	788035	690880	0.8767
9	7	43435	32640	0.7515
9	8	511	256	0.5010
9	9	1	0	0.0000
10	1	1023	1022	0.9990
10	2	174251	173740	0.9971
10	3	6347715	6304280	0.9932
10	4	53743987	52955952	0.9853
10	5	109221651	105911904	0.9697
10	6	53743987	50434240	0.9384
10	7	6347715	5559680	0.8759
10	8	174251	130816	0.7507
10	9	1023	512	0.5005
10	10	1	0	0.0000
11	1	2047	2046	0.9995
11	2	698027	697004	0.9985
11	3	50955971	50781720	0.9966
11	4	866251507	859903792	0.9927
11	5	3548836819	3495092832	0.9849
11	6	3548836819	3439615168	0.9692
11	7	866251507	812507520	0.9380
11	8	50955971	44608256	0.8754
11	9	698027	523776	0.7504
11	10	2047	1024	0.5002
11	11	1	0	0.0000
12	1	4095	4094	0.9998
12	2	2794155	2792108	0.9993
12	3	408345795	407647768	0.9983
12	4	13910980083	13860024112	0.9963
12	5	114429029715	113562778208	0.9924
12	6	230674393235	227125556416	0.9846
12	7	114429029715	110880192896	0.9690
12	8	13910980083	13044728576	0.9377
12	9	408345795	357389824	0.8752
12	10	2794155	2096128	0.7502
12	11	4095	2048	0.5001

4. Коды максимальной мощности

В данном разделе исследуются верхние оценки мощности кодов, обладающих определенными свойствами. Некоторые оценки рассматриваются, в частности, в [4].

4.1. Оценки мощности кодов, исправляющих ошибки данного канала

Пусть дан код $V = \{\mathbf{v_1}, \ldots, \mathbf{v_C}\}$ длины n в алфавите Ω, исправляющий ошибки канала $\Psi(\mathbf{v}) = \{\psi_1(\mathbf{v}), \ldots, \psi_m(\mathbf{v})\}$. Оценим его мощность, $C = |V|$.

Рассмотрим тривиальные примеры.

1) Пусть $m = 1$. В этом случае $\Psi(\mathbf{v}) = \{\psi_1(\mathbf{v})\}$. Разобьем множество Ω^n на классы эквивалентности по следующему отношению:

$$\mathbf{a} \sim \mathbf{b} \quad \Leftrightarrow \quad \psi_1(\mathbf{a}) = \psi_1(\mathbf{b}), \qquad \mathbf{a}, \mathbf{b} \in \Omega^n.$$

В этом случае, набор представителей (по одному представителю) некоторых классов эквивалентности образует код, исправляющий ошибки канала Ψ. При этом, если взять по одному представителю из каждого класса, то будет получен наиболее мощный такой код.

Наибольшая мощность кода, исправляющего ошибки канала Ψ, равна числу различных значений, принимаемых функцией $\psi_1(\mathbf{v})$ на Ω^n. В частности, если $\psi_1(\mathbf{v}) = \mathbf{v}$, то каждый класс эквивалентности состоит из одного элемента и наибольшая мощность кода, исправляющего ошибки канала Ψ равна $|\Omega^n|$, что вполне соответствует содержательному смыслу ситуации: ошибок не происходит.

2) Пусть $m = 2$. В этом случае $\Psi(\mathbf{v}) = \{\psi_1(\mathbf{v}), \psi_2(\mathbf{v})\}$, а таблица декодирования выглядит следующим образом:

$\mathbf{v_1}$	$\mathbf{v_2}$	$\cdots$	$\mathbf{v_C}$
$\psi_1(\mathbf{v_1})$	$\psi_1(\mathbf{v_2})$	$\cdots$	$\psi_1(\mathbf{v_C})$
$\psi_2(\mathbf{v_1})$	$\psi_2(\mathbf{v_2})$	$\cdots$	$\psi_2(\mathbf{v_C})$

Примем ряд дополнительных условий, выполняемых при $i \neq j$:

$$\psi_1(\mathbf{v_i}) \neq \psi_1(\mathbf{v_j}), \qquad \psi_2(\mathbf{v_i}) \neq \psi_2(\mathbf{v_j}), \qquad \psi_1(\mathbf{v_i}) \neq \psi_2(\mathbf{v_j}).$$

Тогда $C \leq \min\{m_1, m_2\}$, где m_1, m_2 — число различных значений, принимаемых функциями $\psi_1(\mathbf{v}), \psi_2(\mathbf{v})$, соответственно.

Вернемся к основной задаче. Код, мощность которого мы оцениваем, исправляет ошибки канала. Это значит, что столбцы в таблице декодирования не пересекаются, откуда следует

Утверждение 18. Пусть V_i — множество различных значений в i-м столбце таблицы декодирования, тогда справедливо неравенство:

$$\sum_{i=1}^{C} |V_i| \leq |\Omega^n|.$$

С другой стороны, элементы каждой из строк таблицы декодирования тоже уникальны. Поэтому

Утверждение 19.

$$C \leq \min_{i \in \overline{1,m}} m_{\psi_i}(\Omega^n),$$

где $m_{\psi_i}(\Omega^n)$ — число различных значений словарной функции $\psi_i(\mathbf{v})$, принимаемых на множестве Ω^n.

Например, рассмотрим двоичный канал с транспозицией не более одной пары соседних букв. Пусть $|V_i|$ — количество различных значений в i-м столбце таблицы декодирования, то есть число слов, которое можно получить из данного описанными транспозициями, — в точности равно числу серий (константных последовательностей) слова $\mathbf{v_i}$. Это следует из того, что транспозиция пары внутри серии не меняет слово. Формально, если кодовое слово представимо в виде $\mathbf{v_i} = \gamma^{p_1}\gamma^{p_2}\ldots\gamma^{p_r}$, где $p_r \geq 1$, то $|V_i| = r$. Например, для вектора (101100) справедливо $|V_i| = 4$. Заметим, что $1 \leq |V_i| \leq n$, что можно показать соответствующими значениями для векторов $(00\ldots0)$ и $(1010\ldots10)$.

4.2. Оценки мощности кодов с данным кодовым расстоянием

В двоичном канале, искажающем не более t букв, словарные функции имеют вид

$$\psi_{\mathbf{a}}(\mathbf{v}) = \mathbf{v} \oplus \mathbf{a}, \qquad \mathbf{a} \in S_t(\mathbf{0}).$$

При этом число различных значений каждой из этих функций не зависит от $\mathbf{a}$, так как $B^n \oplus \mathbf{a} = B^n$. Учтя $m = |S_t(\mathbf{0})| = \sum_{i=0}^{t} \binom{n}{i}$ в неравенстве последнего утверждения, получим *неравенство Хэмминга* (*неравенство сферической упаковки*):

$$C \leq \frac{2^n}{\sum_{i=0}^{t} \binom{n}{i}}.$$

Коды, в которых эта граница достигается, называются *совершенными кодами*. Приведем пример двоичного совершенного кода. Двоичный код Хэмминга [1, параграф 1.7] длины $n = 2^l - 1$, $l \geq 2$, задается матрицей над $\mathrm{GF}(2)$, столбцы которой есть в точности все ненулевые столбцы длины l над $\mathrm{GF}(2)$, то есть матрицей вида

$$H_{l\times n} = \begin{pmatrix} 1 & 0 & 1 & & 0\ldots & 0 & 1 \\ 0 & 1 & 1 & 0 & \ldots & 1 & 1 \\ 0 & 0 & 0 & 1 & \ldots & 1 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \\ 0 & 0 & 0 & 0 & \ldots & 1 & 1 \end{pmatrix}.$$

Нетрудно показать его совершенность, $C = \frac{2^n}{\sum_{i=0}^{n}\binom{n}{i}}$. Этот код исправляет одну ошибку, а умножением проверочной матрицы H на искаженное слово мы получаем вектор, который является двоичной записью номера искаженной буквы.

С другой стороны, исправление кодом V не более t ошибок эквивалентно неравенству $d(V) > 2t$. Поэтому задачу можно сформулировать так: какова максимальная мощность $A_2(n, d)$ двоичного кода с кодовым расстоянием d?

Утверждение 20 (Плоткин)**.** Верны импликации [1, параграф 2.2]:

1. Если d — четное число и $2d > n$, то $A_2(n, d) \leq 2\left\lfloor \frac{d}{2d-n} \right\rfloor$.
2. Если d — нечётное число и $2d + 1 > n$, то $A_2(n,\ d) \leq 2\left\lfloor \frac{d+1}{2d+1-n} \right\rfloor$.
3. Если d — четное число, то $A_2(2d, d) \leq 4d$.
4. Если d — нечетное число, то $A_2(2d + 1, d) \leq 4d + 4$.

4.3. Случай группового канала

Каждая из словарных функций $\psi_k(\mathbf{v})$ канала $\Psi(\mathbf{v})$ является преобразованием множества слов Ω^*. Рассмотрим частный случай, когда канал

$$\Psi(\mathbf{v}) = \{\psi_1(\mathbf{v}), \ldots, \psi_m(\mathbf{v})\}$$

замкнут относительно композиции, то есть образует группу G относительно данной операции.

При этом G разбивает Ω^* на транзитивные множества (классы толерантности)

$$R_{\mathbf{a}} = \{\psi_i(\mathbf{a}), \quad \psi_i \in G\}, \qquad \mathbf{a} \in \Omega^*.$$

Согласно рассуждениям в начале работы, код V наибольшей мощности, исправляющий ошибки данного канала, содержит по одному представителю каждого такого класса $R_{\mathbf{a}}$. Мощность этого кода равна числу транзитивных множеств в Ω^* и определяется с помощью леммы Бернсайда [15].

Рассмотрим пример. Пусть B_q — q-ичный алфавит и $G = \{g_1, \ldots, g_n\}$ — группа циклических сдвигов слов длины n в алфавите B_q, то есть

$$g_i(a_1 \ldots a_{n-i}a_{n-i+1}a_n) = a_{n-i+1} \ldots a_n a_1 \ldots a_{n-i}, \qquad i = \overline{1, n}.$$

Таким образом G — циклическая группа порядка n, которая действует на множестве B_q^n и имеет мощность $|B_q^n| = q^n$.

Пусть теперь n — простое число, тогда искомые транзитивные множества устроены следующим образом. Каждое из q слов, в которых встречается единственная буква,

порождает множество транзитивности мощности 1. Любое другое из $q^n - q$ слово порождает транзитивное множество мощности n. Таким образом число транзитивных множеств r_n равно следующей величине:

$$r_n = \frac{q^n - q}{n} + q.$$

Код наибольшей мощности, исправляющий ошибки данного канала, имеет мощность r_n.

Если n составное, то нахождение числа транзитивных множеств сложнее, но аналогично.

4.4. Коды, стойкие к дезинформации

В данном разделе будет рассмотрена модель, отличная от классической модели теории кодов, исправляющих ошибки. Пусть имеется код $V = \{\mathbf{v_1}, \ldots, \mathbf{v_C}\}$.

Определение 14. Для кодового слова $\mathbf{v}$ кода V в алфавите Ω определим *запретное множество*, или *угрозу*, как произвольное подмножество

$$\mathrm{Vor}(\mathbf{v}, V, \Omega^*) \subseteq \Omega^*.$$

Если V и Ω^* фиксированы, то будем применять обозначение $\mathrm{Vor}(\mathbf{v})$.

Определение 15. Код V называется *стойким к угрозе* $\mathrm{Vor}(\mathbf{v}, V, \Omega^*)$ *в канале* $\Psi(\mathbf{a}) = \{\psi_1(\mathbf{a}), \ldots, \psi_m(\mathbf{a})\}$, если искаженные кодовые слова не входят в запретное множество:

$$\forall\, \mathbf{v} \in V \quad \{\psi_1(\mathbf{v}), \ldots, \psi_m(\mathbf{v})\} \cap \mathrm{Vor}(\mathbf{v}) = \varnothing.$$

Например, код, исправляющий t ошибок, определяется как код, стойкий к угрозам

$$\mathrm{Vor}(\mathbf{v}) = \bigcup_{\mathbf{u} \in V \setminus \mathbf{v}} S_t(\mathbf{u}), \qquad \mathbf{v} \in V.$$

При этом соотношение из определения имеет вид

$$\forall\, \mathbf{v} \in V \quad \{\psi_1(\mathbf{v}), \ldots, \psi_m(\mathbf{v})\} \cap \left(\bigcup_{\mathbf{u} \in V \setminus \mathbf{v}} S_t(\mathbf{u}) \right) = \varnothing.$$

Геометрически это означает, что искаженое слово $\psi_k(\mathbf{v})$ не попадает ни в одну из t-окрестностей слов, отличных от $\mathbf{v}$.

Следующую угрозу назовем *дезинформацией*:

$$\mathrm{Vor}(\mathbf{v}, V, \Omega^*) = \overline{V},$$

где под $\overline{V}$ понимается множество отрицаний кодовых слов V:

$$\overline{V} = \{\overline{\mathbf{v_1}}, \ldots, \overline{\mathbf{v_C}}\}.$$

Код, стойкий к этой угрозе, будем называть *стойким к дезинформации*. Отметим, что данная модель качественно отличается от классической модели кодов, исправляющих ошибки. Если в классической модели мы требуем, чтобы все искажения оказывались «некорректными» (не были кодовыми словами), то здесь постановка противоположная: искажения, приводящие к выходу за пределы кода, являются недопустимыми.

Пусть канал двоичный и искажает не более t букв, тогда код является стойким к дезинформации, если искаженные слова не попадают в множество $\overline{V}$, то есть

$$\forall\, \mathbf{v} \in V, \quad \forall\, \mathbf{a} \in S_t(\mathbf{0}) \quad \psi_{\mathbf{a}}(\mathbf{v}) = \mathbf{v} \oplus \mathbf{a} \notin \overline{V}.$$

Критерием этого является неравенство $\rho(V, \overline{V}) \geq t + 1$, где под расстоянием понимается расстояние Хаусдорфа, то есть

$$\forall\, \mathbf{u}, \mathbf{v} \in V \quad d(\mathbf{u}, \overline{\mathbf{v}}) \geq t + 1,$$

Учитывая равенство $d(\mathbf{u}, \overline{\mathbf{v}}) = d(\mathbf{u}, \mathbf{v} \oplus \mathbf{1}) = |\mathbf{u} + \mathbf{v} + \mathbf{1}| = n - d(\mathbf{u}, \mathbf{v})$, получаем

$$\forall\, \mathbf{u}, \mathbf{v} \in V \quad d(\mathbf{u}, \mathbf{v}) \leq n - (t + 1).$$

В частности, пусть $n \in \mathbb{N}$ — нечетное, а

$$F(\mathbf{a}) = \begin{cases} 1, & \text{если } \|\mathbf{a}\| \leq \frac{n-1}{2} \\ 0, & \text{если } \|\mathbf{a}\| > \frac{n-1}{2} \end{cases}, \qquad \mathbf{a} \in \Omega^n.$$

Тогда код $V = N_F = \{\, \mathbf{a} \in \Omega^n \colon\ \|\mathbf{a}\| \leq \frac{n-1}{2} \,\}$ является стойким к дезинформации в канале с единственной ошибкой, так как справедлива импликация

$$\mathbf{a} \in V,\ \mathbf{e} \in \Omega^n \colon\ \|\mathbf{e}\| = 1 \quad \Rightarrow \quad \|\mathbf{a} + \mathbf{e}\| \leq \frac{n-1}{2},$$

и $\psi(\mathbf{a}) = \mathbf{a} \oplus \mathbf{e} \in V = N_F$, откуда $\psi(\mathbf{a}) \notin \overline{V}$, ведь $V \cap \overline{V} = \varnothing$.

Вернемся к основной задаче данного раздела и оценим максимальную мощность кода, стойкого к дезинформации в канале с t ошибками. Необходимо выполнение неравенства

$$\forall\, \mathbf{u}, \mathbf{v} \in V \quad d(\mathbf{u}, \mathbf{v}) < n - t.$$

То есть необходимо, чтобы все расстояния кода были не больше некоторого числа D.

Определение 16. *Максимальным расстоянием кода* (мощности не менее 2) называется максимальное расстояние между его кодовыми словами:

$$D = D(V) = \max\, \{\, d(\mathbf{u}, \mathbf{v}) \colon\ \mathbf{u}, \mathbf{v} \in V \,\}.$$

Утверждение 21. Пусть $A_2(n, d)$ — максимальная мощность двоичного кода длины n с кодовым (минимальным) расстоянием d, $B_2(n, D)$ — максимальная мощность двоичного кода длины n с максимальным расстоянием D, Тогда при $2t < n$ справедливо неравенство:

$$A_2(n, 2t + 1) B_2(n, 2t) \leq 2^n.$$

Доказательство см. в [5].

Утверждение 22. Существует код мощности 2^D.

Доказательство. Для построения данного кода достаточно взять всевозможные векторы длины n, в которых первые $n - D$ букв совпадают. □

Обратимся теперь к линейным двоичным кодам.

Утверждение 23. Максимальное расстояние линейного кода M определяется максимальным весом среди его слов:

$$D(M) = \max \{ \|\mathbf{v}\| : \mathbf{v} \in M \}.$$

Утверждение 24. Существует линейный двоичный код мощности 2^D.

Утверждение 25. Пусть линейный двоичный код M обладает максимальным расстоянием D, тогда справедливо неравенство:

$$|M| \leq 2^D.$$

Доказательство. Рассмотрим порождающую матрицу кода M в стандартном виде:

$$G_0 = (E_{k\times k}, \overline{H}^T)_{k\times n},$$

Сложив строки этой матрицы, получим вектор

$$\mathbf{v} = (1, 1, \ldots, 1, p_1, \ldots, p_k) \in V,$$

который является кодовым словом, и в котором как минимум k единиц. Тогда имеем $D \geq k$, откуда $2^D \geq 2^k$. Но из линейности кода M его мощность $|M| = 2^k$, и, окончательно,

$$2^D \geq 2^k = |M|,$$

что и требовалось доказать. □

Доказанное неравенство показывает нецелесообразность использования линейных кодов как кодов, стойких к дезинформации, так как их максимальное расстояние D должно быть достаточно малым.

5. Заключение

Были рассмотрены основные вопросы теории кодирования с исправлением ошибок. Рассмотрено понятие дезинформации, а также две математические модели, связанные с этим понятием.

Первая модель связана с классической постановкой задачи обнаружения и исправления ошибок при передаче по каналу связи. В данной работе рассмотрены основные каналы, их описания и формализация. В частности, рассмотрены каналы с инвертациями слов. Для передачи двоичной информации по таким каналам предложены коды, не допускающие инвертации. Найден критерий линейности таких кодов. Получены оценки их мощности, числа (доли), эти данные проверены с помощью ЭВМ. Получен критерий того, что линейный двоичный код, не допускающий инвертации, является МДР-кодом.

Вторая модель описывает ситуацию, когда дезинформация является вовсе недопустимой. Вводится понятие кода, стойкого к дезинформации в данном канале. Проводится оценка максимальной мощности такого кода при дополнительных условиях (например, линейности). На основе этого, в частности, делается вывод о нецелесообразности требования свойства линейности для кода, стойкого к дезинформации в данном канале.

6. Литература

1. Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А. — Теория кодов, исправляющих ошибки — М.: Связь, 1979. — 744 с.
2. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. — М.: Мир, 1976. — 593 с.
3. Нечаев А. А., Линейные коды и полилинейные рекурренты. Спецкурс. — М.: 2006, — 139 с.
4. Дельсарт Ф., Четыре основных параметра кода и их комбинаторное значение. — М.: Кибернетический сборник. №14, с.67-79. Мир, 1977.
5. Леонтьев В. К. Теория кодирования. — М.: Знание, 1977. — 64 с.
6. Леонтьев В. К., Мовсисян Г. Л., Маргарян Ж. Г. Коды в аддитивных каналах. — Доклады Национальной академии наук Армении. Т. 110, № 4, 2010, с. 334-339.
7. Касами Т., Токура Н., Ивадари Ё., Ииагаки Я. — Теория кодирования. — М.: Мир, 1978. — 572 с.
8. Галлагер Р. — Теория информации и надёжная связь. — М.: Советское радио, 1974. — 720 с.
9. Guerrini E., Sala M., A classification of MDS binary systematic codes. — 6 с.
10. Kenneth S. J. K., Construction of Binary Codes. — National University of Singapore, 1999. — 53 с.
11. Шеннон К., Работы по теории информации и кибернетике. — М.: Издательство иностранной литературы, 1963. — 832 с.
12. Леонтьев В. К. Кодирование с обнаружением ошибок. — Проблемы дискретной математики. 1972, Т. 8, № 2, с. 6-14.
13. Аршинов М. Н., Садовский Л.Е. Коды и математика (рассказы о кодировании). — М.: Наука, Главная редакция физико-математической литературы, 1983. — 144 с.
14. Сачков В. Н., Введение в комбинаторные методы дискретной математики. — М.: Наука. Главная редакция физико-математической литературы, 1982. — 384 с.
15. Глухов М. М., Нечаев А. А., Елизаров В. П. Алгебра. Учебник для вузов. — М.: Гелиос. 2003, Т. 1. — 336 с.