МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

В. К. Леонтьев, Э.Н. Гордеев

КОМБИНАТОРНЫЕ АСПЕКТЫ ТЕОРИИ ИНФОРМАЦИИ.

МОСКВА
МФТИ
2019

Представленный текст предназначен для студентов второго курса ФУПМ, имеющих в учебной программе предмет «Дополнительные главы дискретного анализа». Он содержит все необходимые определения, леммы, теоремы и формулы, которые использует преподаватель, однако комментарии, смысл и, зачастую, сущность задач и вопросов, на усвоение и прояснение которых данная дисциплина направлена, остались за рамками текста.

Поэтому данное пособие не может заменить самих лекций. Оно предназначено для следующих целей:

1. Облегчить конспектирование лекций в аудитории, дать возможность больше сосредоточиться на самой лекции, а не на ее конспекте.
2. Часть материала (изложена мелким шрифтом) может быть не включена в лекции из-за недостатка времени, но для любознательных студентов весьма полезна.
3. Так как по данному предмету предусмотрено одно задание, которое сдается на первой неделе мая, в то время как лекции завершаются на 2-3 недели позже, то в данном пособии можно найти необходимый материал для решения задач.
4. Большинство приведенных в тексте упражнений – это задачи из задания, что должно облегчить его сдачу, а также подготовку к контрольной работе.

В силу особенностей учебной программы до третьего курса студентам не читалась «Теория вероятностей», а в данном тексте слово «вероятность» встречается. Однако, кроме самых простых и первичных понятий, которые на лекциях поясняются, из самой «Теории вероятностей» ничего не используется.

Оглавление

1. Лекция 1.

1.1 Часть I. Введение.

1.1.1 Тематика курса.

Очень упрощенно тематику курса можно представить в виде такой схемы.

Дано множество W (множество элементов, множество объектов, множество слов и т.п.; упорядоченное множество, пространство, метрическое пространство и т.п.). Задано свойство S, которым обладает часть элементов этого множества W_s. Требуется найти значения $|W_s|$, оценки значения $|W_s|$, оценки, значение или асимптотику отношения $|W_s|/|W|$и т.п.

Вы скажете, что это чисто комбинаторная задача, и в предыдущих курсах по дискретному анализу (а также в других) вы только этим и занимались. На это есть два возражения: конструктивное и неконструктивное.

Неконструктивное. Название нашего курса содержит слова «Дополнительные главы …». Это значит, что мы будем брать темы, до которых в предыдущих курсах руки не дошли.

Конструктивное. Обозначим некоторые особенности, акценты, выделяющие тематику курса. Такие особенности в предыдущих и текущих курсах дискретного анализа задавались, например, следующими понятиями: булевы функции, алгебра логики, алгебраические конструкции (высшая алгебра), теория алгоритмов, формальные системы, комбинаторные методы и пр.

Ключевым объектом исследования у нас будет понятие *«слово»*. Понятие *слово* тесно связано с понятием *информация*, а понятие *информация* – с понятием *алгоритм*.

Понять, что такое «информация», пока не удалось никому. Однако многочисленные исследования и рассуждения на тему этого феномена позволяют предположить, что тут есть определённая проблема, находящаяся на стыке математики и других естественных и гуманитарных дисциплин.

Принято считать, что информация «локализует» объект и что чем больше информации об объекте, тем точнее его можно определить.

Не менее трудно понять, что такое «передача информации», если не ограничиваться классической моделью Шеннона и попытаться найти адекватные выражения и описания для значительно более общих ситуаций. В частности, передача информации и перенос информации могут означать разные деяния.

Причинно-следственные связи являются важнейшим компонентом любой науки, но не имеют сколько-нибудь серьёзного фундамента за пределами ряда разделов точных наук.

Одним из способов выражения такого рода связей является понятие «зависимости». Такая зависимость может быть «функциональной», «линейной», «вероятностной» и так далее.

Понятие информации даёт возможность ввести в обиход «информационную» зависимость, выражающую в какой-то степени один из видов причинно-следственных связей, но не совпадающих ни с одним из приведённых выше видов зависимостей.

Под анализом информации обычно подразумевают извлечение полезных данных из «текста», в котором эти данные представлены косвенным, искажённым или недостаточно полным образом. Такой анализ необходим во многих содержательных ситуациях, связанных с восстановлением повреждённых сообщений, сравнением генетических последовательностей, автоматическим переводом с одного языка на другой и так далее.

Ключевые понятия в такой деятельности следующие: текст, контекст, слово, фрагмент, подслово, смысл, форма, содержание, близость, похожесть и так далее.

Под анализом понимается изучение изучение возможностей восстановления исходной информации с максимальной достоверностью.

Исходным объектом является слово. Это слово представимо некоторым набором своих частей. Требуется путём анализа этих частей восстановить исходное слово с той точностью, которую допускает исходная информация. Под «точностью» в содержательном смысле понимается множество всех кандидатов на роль неизвестного слова. Под контекстом понимается априорная информация или информация о множестве слов, которому заведомо принадлежит предмет поиска. Априорная информация может иметь различную форму: предикатное описание, «таблица обучения», набор признаков, геометрическое расположение и так далее.

Например, объекты из упомянутого выше множества W надо как-то задавать, *кодировать*. Здесь возникают *слова* как *коды* объектов. Значения и оценки $|W_s|$, упомянутые выше, надо как-то получать. Один из самых распространенных приемов – сконструировать процедуру порождения объектов из W_s и посмотреть, сколько объектов она породит. Здесь и возникает *алгоритм*.

1.1.2 Слово, информация, алгоритм.

Два интуитивных понятия: *информация* и *алгоритм* взаимно обуславливают друг друга.

Формальные подходы к понятию алгоритма заслужили внимание благодаря гипотезам типа ***тезиса Черча***.

Этот тезис говорит об эквивалентности широкого неформального и смутного понятия "интуитивный алгоритм" узкому и весьма замысловатому, на первый взгляд, формализму типа *алгорифма Маркова* или *машины Тьюринга (МТ)*. Утверждается, что для любой задачи, для которой существует «интуитивный» алгоритм решения, можно, например, построить МТ, которая будет решать эту задачу.

Сделаем одно замечание. Во всех перечисленных ниже формальных подходах алгоритм имеет дело с преобразованием одних слов в другие. То есть, условие задачи Z, которую решает алгоритм, можно рассматривать как слово в некотором алфавите A.

Обозначим все множество слов этого алфавита через A*. Подмножество слов алфавита назовем *языком*. Обратим внимание, что в содержательном смысле не все слова алфавита являются условиями индивидуальных задач из Z.

Очевидно, что *алгоритма* без *информации* не существует. Если мы хотим дать определение *алгоритма*, нам нужно определение *информации*.

Следующая фраза, конечно, не является математическим определением (а такового, видимо, и не существует для «интуитивно очевидных понятий» типа *информации, числа, алгоритма* и пр.), но кое-что проясняет.

Информация – это продукт деятельности субъекта (человека), предназначенный для другого субъекта (человека), представляющий собой инструкцию (часть инструкции) по преобразованию окружающего мира с минимальными затратами ресурсов (энергии).

Но, как и для *алгоритма*, для *информации* существует тезис, в каком-то смысле «заменяющий» определение.

Тезис. Любая *информация* может быть представлена *словом* в конечном алфавите.

Справедливо и обратное: *информации* без *алгоритма* не существует.

По поводу этих тезисов заметим, что они справедливы только в одну сторону. Не любая *Машина Тьюринга* представляет собой *алгоритм* и не любое *слово* является *информацией*!

Итак, условие задачи – это *информация*, текст алгоритма – *информация*, результат работы алгоритма – *информация* и т.п. Еще раз хочу подчеркнуть, что нет *алгоритма* без *информации* и наоборот.

Теперь поясним, что будет пониматься под *количеством информации*, *кодированием информации* и *сжатием информации*.

Очевидно, что множество слов может быть представлено одним словом. (Например, с помощью операции *конкатенации*.)

Опр. Количество символов в слове α будем называть *длиной слова* α и обозначать через l(α) или |α|.

Опр. Конкатенация слов α и β получается путем последовательной записи символов одного слова вслед за символами другого, т.е.

$$\begin{cases} \alpha = b_1, b_2, b_3, \ldots, b_k \\ \beta = j_1, j_2, j_3, \ldots, j_S \end{cases} \Rightarrow \alpha\beta = b_1 b_2 b_3 \ldots b_k j_1 j_2 j_3 \ldots j_S$$

Поэтому понятия *кодирование информации*, *количества информации* и *сжатие информации* могут относиться как к одному слову, так и к множеству слов.

Кодирование информации – это создание слова α в алфавите А.

Так как *информация словом* не является, а лишь представляется, а не любое *слово* – это *информация*, то *слово,* представляющее собой *информацию* (об объекте) будем называть *информационным объектом*.

Количество информации – это некоторая количественная характеристика H(α), сопоставляемая слову (множеству слов) α.

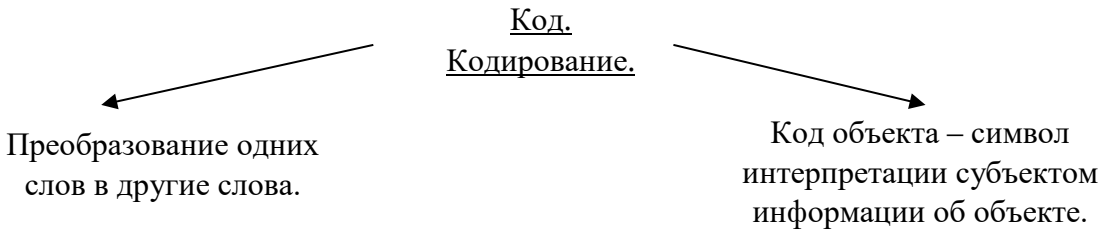
Сжатие информации - это преобразование одного слова α в другое слово β. Поэтому сжатие – это преобразование информации, которое можно задать с помощью некоторого отображения φ. Таким образом, β=φ(α).

Вообще говоря, термин *сжатие информации* тесно связан с понятием *количества информации*. Сжатие происходит тогда, когда количество информации, приходящейся на один символ исходного слова α меньше, чем количество информации, приходящейся на один символ слова β=φ(α).

Понятие *кодирование* применительно к информации будет рассматриваться в двух аспектах:

Создание самого *Информационного объекта*. То есть нанесение на материальный носитель *информации* о некотором *объекте* в виде слова α в алфавите А. В этом случае данное *слово* будет рассматриваться как *код объекта*. То есть *код объекта* – это символ интерпретации субъектом *информации об объекте*. (Назовем это *кодированием в широком аспекте*.)

Преобразование информации. Вначале информация была представлена словом α, а затем **эта же** информация представляется словом β, которое получено из α в результате некоторого преобразования φ. Тогда слово β=φ(α) называется *кодом* слова α. (Назовем это просто *кодированием*.)



Первый аспект возникает тогда, например, в программировании, когда нужно представить объект в виде набора данных. Эта проблема возникает как для «математических объектов»: числа, фигуры, группы, поля, алгоритмы и т.п., так и для объектов из других областей знаний: месторождения в геологии, болезни в медицине, каталоги в промышленности и т.п. Здесь результатом кодирования является слово, в то время как исходный объект кодирования либо представляет собой формализованный математический объект, либо вообще не формализован.

Часто бывает, что кодирование (создание *информационного объекта*) - это итерационный процесс. То есть, сначала есть объект В , есть информация о нём I(B), потом есть информация об информации о нём I(I(B)) и т.д. В результате можно получить некоторую структуру данных, состоящую из всех таких *информаций*.

$$B \to I(B) \to I(I(B)) \to \ldots\ldots \quad КОД$$
$$\Rightarrow I(I(I(B))) - структура\ данных$$

1.1.3 Примеры кодировок

Приведем простые примеры кодирования объектов.

Целые числа

Целое число n может быть представлено (закодировано), например, следующими тремя способами.

а) Словом α = 01…10 (*n* штук «1» и 0 по бокам) в алфавите А=(0,1). В этом случае длина кода l(α) (количество символов алфавита в слове) равна *n+2*;

б) Словом α = $a_1a_2\ldots a_k$ в виде десятичной записи в алфавите А=(0,1,…,9). Длина кода равна [lg*n*]+1.

в) В мультипликативной форме (произведение степеней простых множителей) в виде слова α = a_1^{p1} $a_2^{p2}\ldots a_k^{pk}$, (здесь $a_1\ldots a_k$ – простые числа).

Длина кода: $\sum_{i=1}^{k}(\log a_i + \log p_i)$.

Вектора и матрицы

Вектор с компонентами $a_1 \ldots a_n$, записанными в десятичной форме имеет, может быть представлен в виде слова длины $\lg n + \sum_{i=1}^{n} \log a_i$.

Матрица M с компонентами в десятичной форме, $M = \begin{Vmatrix} a_{11} & \ldots & a_{1n} \\ \ldots & \ldots & \ldots \\ a_{m1} & \ldots & a_{mn} \end{Vmatrix}$, может быть представлена словом длины $\lg n + \lg m + \sum_{i=1}^{m} \sum_{j=1}^{n} \log a_{ij}$.

$G = (V, E)$ – неориентированный граф.

Здесь множество вершин обозначено через V, а множество ребер через E. Пусть $|V| = n$, $|E| = m$.

Граф можно представить, например, следующими тремя способами (все числа в десятичной форме)

а) Списком ребер $\{(i_1, j_1), (i_2, j_2), \ldots, (i_m, j_m)\}$.

б) Списком соседей (вершин), записанных последовательно для вершины 1, вершины 2,…, вершины n.

$$\underset{1}{(i_1^1 \ldots i_{k_1}^1)}, \quad \underset{2}{(i_1^2 \ldots i_{k_2}^2)}, \quad \underset{\ldots}{\ldots}, \quad \underset{n}{(i_1^n \ldots i_{k_n}^n)}$$

в) Матрицей смежности

$A = |\ a_{ij}\ |_{n \times n}$, $a_{ij} = \begin{cases} 1, (i, j) \in E \\ 0, (i, j) \notin E \end{cases}$.

Алфавит A кроме арабских цифр будет включать вспомогательные разделители, например, «,» и «;».

Длина кода l(α) без учета разделителей для каждого из трех случаев следующая:

а) $4n + 10m \le l(\alpha) \le 4n + 10m + (n+2n)$.

б) $2n + 8m \le l(\alpha) \le 2n + 8m + 2\log n$.

в) $l(\alpha) = n^2 - n - 1$.

1.1.4 Свойства кодировок

К *коду* β *объекта* α обычно применяют следующие требования.

Декодируемость. В случае кодирования (как преобразования слов) это свойство не вызывает вопросов. Пусть есть объект α, код $\beta = \varphi(\alpha)$. Декодируемость означает наличие возможности восстанавливать α по коду $\beta = \varphi(\alpha)$. В случае же кодирования в широком аспекте словом является только β. Сам объект α словом не является. Поэтому в каждом конкретном случае нужно определять понятие *восстановление*. В качестве примера можно привести пример присвоения идентификатора (слово в алфавите) неформализованному объекту (изделие, книга, человек). Здесь под восстановлением понимается нахождение конкретного объекта по его идентификатору.

Неизбыточность. Это требование означает стремление уменьшить длину кода. Но пока речь не идет об оптимизации и минимизации. Неизбыточность в широком смысле означает не поиск минимально возможной (по длине кода) кодировки, а использование кодировки без искусственного увеличения ее длины. То есть из слова-кода нельзя выбросить букву без нарушения требования дешифруемости.

Разумность. Это термин из *теории сложности алгоритмов* (см., например, [1], стр. 23-24). Он означает, что способ кодировки должен соответствовать тем целям, для которых информация будет использоваться в дальнейшем.

Например, если мы решаем *задачу о простоте числа* (по заданному натуральному числу N требуется ответить на вопрос, является ли это число простым), то кодировка в мультипликативной форме (см. пример выше) не является *разумной*. Для большинства практических задач первый из видов кодировок целого числа n в приведенном выше примере не является *разумной* кодировкой.

Пусть мы теперь имеем *"разумные" кодировки*. На этом уровне принимается интуитивная гипотеза о некоторой эквивалентности всех возможных "разумных" кодировок входных данных одной и той же массовой задачи.

Второй пример показывает варианты длины кодировок в зависимости от способа задания чисел. Варианты длины кодировок в третьем примере зависят уже от структуры входных данных. Но в обоих случаях можно говорить о, так называемой, полиномиальной эквивалентности.

В дальнейшем нам полезно будет понятие *полиномиальной эквивалентности* двух функций. Две неотрицательные функции S(n) и R(n) полиномиально эквивалентны, если существуют два полинома Q(x) и P(x) такие, что для всех n справедливы неравенства $S(n) \le P(R(n))$ и $R(n) \le Q(S(n))$.

Пусть S и R две "разумные" схемы кодирования задачи массовой Z. Длины входа в этих схемах для задачи I обозначим через S(I) и R(I). К ним можно применить определение полиномиальной эквивалентности.

В примере кодирования графа все три схемы полиномиально эквивалентны.

2.1. Часть II. Кодирование множеств и подмножеств.

2.1.1 Примеры экстремальных задач теории множеств.

Пусть задано n-элементное множество W. Для его задания одного числа n уже достаточно. Предполагается, что его элементы как-то идентифицируются, например числами 1,2,…,n. Но во избежание путаницы идентификаторами обычно служат буквы с индексами, поэтому боле распространенный способ задания множества – это перечисление его элементов в виде: $a_1,a_2,\ldots,a_n$.

Тогда любому подмножеству B множества W можно сопоставить код этого подмножества в виде бинарного слова $X_B=(x_1,\ldots,x_n)$ длины n, где

$$x_i = \begin{cases} 1, a_i \in B \\ 0, a_i \notin B \end{cases}.$$

Следующие задачи хорошо известны благодаря естественности постановок и простоты решения.

Определение. Семейство $S = \{V_1 \dots V_N\}$ подмножеств множества $A = \{a_1 \dots a_n\}$ называется *неразделённым*, если любые два подмножества из S имеют непустое пересечение.

Утверждение. Если S — неразделимое семейство подмножеств из A, то

$$|S| \leq 2^{|A|-1}. \quad (1.1)$$

Доказательство. Если S неразделимо, то следующие две совокупности подмножеств $V_1 \dots V_N$ и $\overline{V_1} \dots \overline{V_N}$ не имеют общих элементов. Отсюда следует неравенство $2N \leq 2^{|A|}$.

Что и требовалось доказать.
Ясно, что оценка (1.1) является достижимой, так как семейство подмножеств, содержащих фиксированный элемент $a_i \in A$, имеет мощность $2^{|A|-1}$.

Замечание. следующий пример показывает, что оценка достижима не только для подмножеств с указанной выше структурой. Пусть $n = 3$ и $S = \{a_1a_2, a_1a_3, a_2a_3, a_1a_2a_3\}$, тогда S неразделимо и $|S| = 4 = 2^{|A|-1}$.

Определение. Семейство $S = \{V_1 \dots V_N\}$ называется *слабо совместимым*, если

$$\left|V_i \cap V_j\right| \leq 1, i \neq j, i, j = \overline{1, N}.$$

Утверждение. Если S — слабо совместимое семейство подмножеств, то справедливо неравенство:

$$|S| \leq 1 + \binom{n}{1} + \binom{n}{2},$$

где $n = |A|$.

Доказательство. Пусть $S = \{V_1 \dots V_N\} \in 2^A$ и S — слабо совместимо. Если S нельзя расширить с сохранением свойства слабой совместимости, то $|V_k| \leq 2$ для $k = 1,2, \dots, N$. Действительно, если $|V_k| = \{a, b, c\}$, то рассмотрим новое семейство $S' = \{V_1, \dots, V_{k-1}, (ab), (bc), V_{k+1}, \dots, V_N\}$. Отметим сначала, что S' слабо совместимо. Действительно, если $\left|V_p \cap (ab)\right| = 2$, то $\left|V_p \cap (abc)\right| \geq 2$ или $\left|V_p \cap V_k\right| \geq 2$, что противоречит исходному предположению о слабой совместимости V. Далее $|S'| = |S| + 1$, что противоречит предположению о «нерасширяемости» множества S. Таким образом, $|V_k| \leq 2$ и, значит, $|S| \leq 1 + \binom{n}{1} + \binom{n}{2}$. Что и требовалось доказать.

Пусть $S = \{V_1 \dots V_N\} \subseteq 2^A$ и $\left|V_i \cap V_j\right| = 1$ при $i \neq j$. Как велико может быть N?

Утверждение. Справедливо неравенство

$$|S| \leq |A|.$$

Доказательство. Если $|A| = n$ и $\lambda(V_k)$ — двоичный вектор, кодирующий V_k, то условие $\left|V_i \cap V_j\right| = 1$ трансформируется следующим образом:

$$\left(\lambda(V_i), \lambda\left(V_j\right)\right) = 1. \quad (1.2)$$

Исходя из (1.2), покажем, что векторы $\lambda(V_1), \dots, \lambda(V_N)$ линейно-независимы. Матрица Грама этой системы векторов имеет следующий вид:

$$C = \begin{Vmatrix} |V_1| & 1 & \dots & 1 \\ 1 & |V_2| & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & |V_N| \end{Vmatrix}.$$

Рассмотрим теперь следующую матрицу A_N с $b_i \geq 0$:

$$A_N = \begin{Vmatrix} b_1 + 1 & 1 & \dots & 1 \\ 1 & b_2 + 1 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & b_N + 1 \end{Vmatrix}.$$

Если $|A_N| = \Delta_N$, то

$$\Delta_N = \begin{vmatrix} b_1 + 1 & 1 & \dots & 1 \\ 1 & b_2 + 1 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & b_N + 1 \end{vmatrix} =$$

$$= \begin{vmatrix} b_1 + 1 & 1 & \dots & 1 \\ 1 & b_2 + 1 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & 1 \end{vmatrix} + \begin{vmatrix} b_1 + 1 & 1 & \dots & 1 \\ 1 & b_2 + 1 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & b_N \end{vmatrix} =$$

$$= \begin{vmatrix} b_1 & 0 & \dots & 1 \\ 0 & b_2 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{vmatrix} + b_N \Delta_{N-1} = b_1 b_2 \dots b_{N-1} + b_N \Delta_{N-1} \quad (1.3)$$

Из (1.3) следует при $b_k \geq 0$

$$\Delta_N = b_1 b_2 \dots b_N \left(1 + \tfrac{1}{b_1} + \dots + \tfrac{1}{b_N}\right),$$

где при $b_i = 0$ из вычёркиваются все переменные b_i.

Утверждение доказано.

Отметим далее, что приведённые выше факты об экстремальных комбинаторных задачах теоретико-множественного содержания составляют незначительную часть всего материала, относящегося к этой предметной области, и служат лишь иллюстрацией характера результатов и методов их получения.

2.1.2 Отображения конечных множеств.

Если A, B — конечные множества, то положим, по определению, что A^B — это множество всех отображений из A в B, то есть соответствий f этого вида:

$$A \overset{f}{\to} B.$$

Каждое такое отображение может быть задано с помощью матрицы

$$C = \begin{pmatrix} a_1 & a_2 & \dots & a_n \\ \gamma_1 & \gamma_2 & \dots & \gamma_n \end{pmatrix},$$

где $A = \{a_1, \dots, a_n\}$, $B = \{b_1, \dots, b_m\}$ и $\gamma_k \in B$ для $k = \overline{1,n}$. Таким образом, первая строка матрицы C — это множество A, а вторая — мультимножество над B. Ясно, что число таких матриц C равно m^n, и отсюда следует формула

$$|A^B| = |B|^{|A|}.$$

Если $A = B$ и f — взаимно однозначное отображение A в себя, то f называется *перестановкой* и стандартная запись для f выглядит следующим образом:

$$f = \begin{pmatrix} a_1 & a_2 & \dots & a_n \\ a_{i_1} & a_{i_2} & \dots & a_{i_n} \end{pmatrix}.$$

Множество S_n, где $n = |A|$, называется *множеством перестановок степени* n.

Как комбинаторный объект множество S_n является неисчерпаемым источником проблем, зачастую имеющих глубокий математический и содержательный смысл.

Если на множество S_n ввести операцию суперпозиции, то S_n превращается в группу, которая называется симметрической группой степени n. Ясно, что $|S_n| = n!$. Группа S_n является универсальным алгебраическим объектом в силу следующего утверждения.

Теорема Кэли. Каждая конечная группа изоморфна некоторой подгруппе S_n.

Каждая перестановка $g \in S_n$ единственным образом разлагается в произведение независимых циклов:

$$g = v_1 v_2 \dots v_m, \quad (1.4)$$

где $v_i = \left(a_1^i a_2^i \dots a_k^i\right)$. Представление (1.4) играет существенную роль во многих проблемах, связанных с группой S_n. Например, по разложению (1.4) легко найти порядок элемента g в группе S_n. Действительно, искомый порядок $t(g)$ равен Н.О.К. длин циклов v_i, то есть

$$t(g) = H.O.K.(|v_1|, |v_2|, \dots, |v_m|).$$

1.3 Упражнения.

Пусть $A = \{a_1, \dots, a_n\}$ — конечное множество, 2^A — семейство всех подмножеств A; $\binom{A}{k}$ — семейство всех k-элементных подмножеств A; «∪», «∩», «—» — обычные теоретико-множественные операции объединения, пересечения и дополнения.

Первым объектом при изучении системы $\{2^A, \cap, \cup\}$ является отображение

$$2^A \to \{0,1,2, \dots\},$$

определяемое следующим образом:

$$B \to |B|,$$

где $|B|$ — число элементов множества B.

Задачи первого уровня сложности, связанного с отображением, часто имеют выражение в терминах биномиальных коэффициентов $\binom{n}{k}$, имеющих «бытовое» название — «число сочетаний изn по k».

По определению

$$\binom{n}{k} \stackrel{def}{=} \frac{n!}{k!(n-k)!},$$

и комбинаторный смысл этой формулы следующий: $\binom{n}{k}$ — это число k-элементных подмножеств n-элементного множества.

Биномиальные коэффициенты в значительной мере представляют собой «базис» элементарной комбинаторики и на их языке формулируются многие факты, представляющие интерес для комбинаторного анализа в целом.

Имеют место следующие хорошо известные и легко проверяемые соотношения:

1. $|2^A| = 2^{|A|}$,

2. $\left|\binom{A}{k}\right| = \binom{|A|}{k}$,

а также простейшие соотношения для биномиальных коэффициентов (б. к.)

1. $\binom{n}{k} = \binom{n}{n-k}$,
2. $\binom{n}{k} = \frac{n}{k}\binom{n-1}{k-1}$,
3. $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$,
4. $\sum_{k=0}^{n}(-1)^k \binom{n}{k} = 0$,
5. $\sum_{r=0}^{k}(-1)^r \binom{n}{r} = (-1)^k \binom{n-1}{k}$,
6. $\sum_{k=0} \binom{N}{m+k}\binom{M}{n-k} = \binom{N+M}{n+m}$. (1.5)

Формулы вида (1.5) составляют часть общего класса комбинаторных тождеств, насчитывающих тысячи соотношений подобного рода и играющих значительную роль во многих разделах математики и теоретической физики. Наиболее естественный способ для обоснования приведённых выше формул и получения новых — это метод производящих функций.

Упражнения.

a) Найти число подмножеств из $\binom{A}{k}$, содержащих фиксированный элемент $a_i \in A$.

b) Найти число подмножеств $V \in \binom{A}{k}$, которые пересекаются с данным подмножеством $M \in 2^A$.

c) Найти число решений уравнения

$$x \cup y = A, \qquad x, y \in 2^A,$$

при $(x, y) \neq (y, x)$ и для упорядоченных решений.

d) Найти число решений уравнения

$$x \cap y = a \text{ где } x, \qquad y, a \in 2^A$$

при $(x, y) \neq (y, x)$ и для упорядоченных решений.

e) Найти число неупорядоченных решений уравнения

$$x_1 \cup x_2 \cup \ldots \cup x_k = A$$

при $x_i \in 2^A$, где $|A| = n$.

f) Найти число решений «неравенства» $a \subseteq x \subseteq b$.

g) Найти число подмножеств мощности k, которые пересекаются с данным множеством $M \subseteq A$, где $|A| = n$, $|M| = m$.

2. Лекция 2. Упорядоченные множества.

2.1 Порядок на множествах.

Если перейти от «обычного» конечного множества к частично упорядоченному

множеству (Ч.У.М.), то количество и сложность комбинаторных проблем резко возрастает. Понятие упорядоченности (что больше?) играет исключительно важную роль во многих разделах математической науки и, в том числе, в комбинаторном анализе. Поэтому многие из задач, относящихся к Ч.У.М., имеют глубокий математический и содержательный смысл.

Итак, пусть $(X, \leq)$ — множество с бинарным отношением $(\leq)$, удовлетворяющим стандартным требованиям рефлексивности, антисимметричности и транзитивности. Такое множество называется частично упорядоченным или кратко (Ч.У.М.).

Функция $\zeta(x, y)$, определённая на Ч.У.М. $(X, \leq)$ соотношением

$$\zeta(x, y) = \begin{cases} 1, & \text{если } x \leq y, \\ 0, & \text{если не так.} \end{cases} \quad (2.1)$$

характеризует «сравнимость» элементов $(X, \leq)$ и называется *дзета-функцией*.

Если X конечно, то соотношение (2.1) характеризует инцидентность элементов X и может быть описано с помощью соответствующей матрицы $A = ||\zeta(x, y)||$, которая и называется матрицей инцидентности Ч.У.М. $(X, \leq)$.

Нетрудно понять, что A является верхне-треугольной и потому существует обратная матрица $A^{-1} = ||\mu(x, y)||$, элементы которой $\mu(x, y)$ определяют *функцию Мёбиуса* частичного порядка $(X, \leq)$.

Любое подмножество $V \subseteq X$ такое, что элементы V попарно сравнимы, образует *цепь*, то есть $V = \{x_{i_1} < x_{i_2} < \cdots < x_{i_k}\}$.

Антицепь — это множество из $(X, \leq)$ с противоположным свойством, то есть любые два элемента антицепи V' несравнимы или

$$\zeta(x, y) = 0 \text{ для } x, y \in V'.$$

Цепи и антицепи — это подмножества Ч.У.М., в значительной степени характеризующие его структуру, и потому им обычно уделяется внимание при исследовании конкретных Ч.У.М. Одним из самых значительных результатов в теории Ч.У.М., касающихся введённых выше понятий, является следующее утверждение.

Теорема (Дилуорс). Максимальная мощность антицепи в $(X, \leq)$ равна минимальному числу непересекающихся цепей, покрывающих X.

Доказательство. В одну сторону очевидно: если порядок разбит на k непересекающихся цепей, то любая антицепь пересекается с каждой из цепей не более чем по одному элементу и в антицепи не больше k элементов.

В другую сторону - индукция по числу элементов в порядке.

База — порядок из одного элемента — очевидно выполнена.

Пусть утверждение теоремы справедливо для порядков с количеством элементов не больше n.

Рассмотрим порядок P, в котором n+1 элемент. В любом конечном порядке есть минимальные элементы. Пусть m —минимальный элемент в порядке P. Отбрасывая этот элемент, получаем порядок Q, для которого утверждение теоремы выполнено. Обозначим наибольший размер антицепи в Q через s.

По предположению индукции порядок Q можно разбить на s непересекающихся цепей. Наибольший размер антицепи в P либо равен s, либо равен s + 1.

В последнем случае m содержится в антицепи размера s + 1 и порядок P легко разбивается на s + 1 цепь: одна состоит только из элемента m, а остальные разбивают Q на s непересекающихся цепей.

Осталось рассмотреть случай, когда наибольший размер антицепи в P равен s. Элемент m тогда сравним с какими-то элементами порядка Q, а поскольку он минимальный, то он меньше каких-то элементов. Рассмотрим в каждой цепи в Q минимальный элемент, входящий в антицепь максимального размера. Совокупность всех этих элементов образует антицепь M. Действительно, если для двух таких элементов a < b, то рассмотрим максимальную антицепь, в которой лежит a. Эта антицепь пересекается с цепью, в которой лежит b.

В силу минимальности b по транзитивности получаем сравнимость двух элементов антицепи, что дает противоречие. Один из элементов q построенной антицепи M сравним с m (иначе в P есть антицепь размера s+1).

То есть, получается, что m < q, а все элементы, меньшие q в цепи C разбиения Q на s непересекающихся цепей, не входят в антицепи размера s. Выделим цепь, состоящую из m и всех элементов цепи C, начиная с q и больше. Порядок на оставшихся элементах не содержит антицепей размера s (так как любая такая антицепь обязана пересекать остаток от цепи C) и в нем не больше n элементов. Значит, для этого порядка утверждение теоремы справедливо и его можно разбить на s−1 непересекающуюся цепь. Добавляя выделенную цепь, получаем разбиение исходного порядка P на s цепей.

Таким образом, утверждение теоремы справедливо и для порядка P. По принципу математической индукции теорема справедлива для всех конечных порядков.

Теорема доказана.

Мы используем эту теорему для получения следующей известной верхней границы числа антицепей в произвольном конечном Ч.У.М.

Утверждение. Если m — максимальная мощность антицепи в X, то для числа $N(x)$-антицепей в X справедлива оценка

$$N(x) \leq \left(1 + \frac{|x|}{m}\right)^m.$$

Доказательство. В силу теоремы Дилуорса для X справедливо следующее разложение на множество непересекающихся цепей:

$$X = \bigcup_{k=1}^{m} S_k.$$

Каждая антицепь в X содержит не более одного элемента из каждой цепи S_k. Отсюда получаем границу

$$N(x) \leq \prod_{k=1}^{m}(1 + |S_k|) \leq \left(\frac{\sum_{k=1}^{m}|S_k| + 1}{m}\right)^m = \left(1 + \frac{|x|}{m}\right)^m.$$

Здесь мы использовали неравенство о среднем арифметическом и среднем геометрическом и дизъюнктивность S_k, то есть

$$|x| = \sum_{k=1}^{m}|S_k|.$$

Мощность максимальной антицепи в $(B^n, \leq)$ оценивается в следующем утверждении, которое называется леммой Шпернера.

Лемма Шпернера. Если V — антицепь в $(B^n, \leq)$, то

$$|V| \leq \binom{n}{\left[\frac{n}{2}\right]}.$$

Доказательство. Общее число цепей длины n, «соединяющих» точки 0^n и 1^n, равно $n!$.

Если $V = \{a_1 a_2 \dots a_k\}$ — антицепь, то число цепей длины n, проходящих через a_r, равно $||a_r||!\,(n - ||a_r||)!$, и так как все эти цепи различны, то общее число цепей, проходящих через V, оценивается как

$$\sum_{a_r \in V} ||a_r||!\,(n - ||a_r||)! \le n!.$$

Отсюда получаем

$$\sum_{a_r \in V} \binom{n}{||a_r||} \le 1. \quad (2.2)$$

Так как $\binom{n}{k} \le \binom{n}{[\frac{n}{2}]}$, то из (2.2) следует

$$|V| \le \binom{n}{[\frac{n}{2}]},$$

что и требовалось доказать.

Так как любой слой B_k^n в B^n является антицепью, то лемму Шпернера можно уточнить следующим образом: если n — чётное число, то единственной антицепью мощности $\binom{n}{[\frac{n}{2}]}$ является «слой». Если же n нечётно, то таких антицепей ровно две — это «слои» $B^n_{\frac{n-1}{2}}$, $B^n_{\frac{n+1}{2}}$.

Пусть Z^+ — множество натуральных чисел со следующим бинарным отношением: $\frac{x}{y}$ — x — делитель y. Это отношение порождает отношение частичного порядка на Z^+:

$$x \le y \rightleftharpoons x/y.$$

Множество $(Z, \le)$ с этим отношением порождает дистрибутивную решётку в том смысле, что для любой пары натуральных чисел существует единственный минимальный (в смысле введённого порядка) и единственный максимальный элемент. Этими элементами являются классические теоретико-числовые функции: наибольший общий делитель и наименьшее общее кратное. Если же рассмотреть только ограниченный интервал $I_n = [1, n]$ натурального ряда с этим же отношением частичного порядка, то множество $(I_n, \le)$ будет также Ч.У.М., но уже не будет решёткой. Если [a, b] — интервал в $(I_n, \le)$, то «длина» этого интервала равна $\tau\left(\frac{b}{a}\right)$, то есть

$$|[a, b]| = \tau\left(\frac{b}{a}\right),$$

где $\tau(m)$ —теоретико-числовая функция — число всех делителей m. Если [a, b] = [3, 36], то $|[3{,}36]| = |3{,}6{,}9{,}12{,}18{,}36| = 6 = \tau\left(\frac{36}{3}\right) = \tau(12)$. Если [a, b] = [12, 48], то $|[12{,}48]| = |12{,}24{,}48| = 3 = \tau\left(\frac{48}{12}\right) = \tau(4)$.

Оценим теперь мощность максимальной антицепи в $(I_n, \le)$. Пусть $A = \{1 \le a_1 < a_2 < \dots < a_k \le n\}$ — произвольное подмножество I_n.

Определение. Преобразование $T(A) = \{2a_1, a_2, \dots, a_k\}$ — называется *сдвигом* A.

Лемма. Если A — антицепь в I_n и $a_1 < \frac{n}{2}$, то $T(A)$ — антицепь в I_n.

Доказательство. С одной стороны, ни одно из чисел a_i не делится на $2a_1$. С другой, если $\frac{2a_1}{a_r} = \alpha$ — целое число, то из условий $\frac{2a_1}{a_r} \ge 1$ и $\frac{2a_1}{a_r} < 2$ следует, что $\frac{2a_1}{a_r} = 1$ или a_1/a_r — что противоречит тому, что A — антицепь. Таким образом, $T(A)$ — антицепь.

Следствие. Если A — антицепь в I_n, то $|A|A \le \left[\frac{n}{2}\right] + 1$.

Доказательство. Из предыдущей леммы следует, что если A — антицепь в I_n, то A может быть «размещена» в интервале $\left\{\left[\frac{n}{2}\right]+1, \dots, n\right\}$, что и завершает обоснование приведённой выше верхней граница. Само «размещение» A проводится путём применения итераций преобразования T.

Замечание. Иногда это следствие формулируют в следующей теоретико-числовой форме: если в отрезке [1, n] выбрать $\left[\frac{n}{2}\right]+1$ число, то найдётся пара из них таких, что одно делится на другое.

2.2 ЧУМ подслов и фрагментов.

Пусть $A = \{a_1 \dots a_n\}$ — конечный алфавит и A^* — множество всех слов конечной длины над алфавитом A. Под конкатенацией (произведением) слов $a, b \in A^*$ будем понимать слово ab. При этом удобно ввести понятие пустого слова Λ, которое обладает следующим свойством:

$$a\Lambda = \Lambda a$$

для любого слова $a \in A^*$.

Множество $(A^*,\cdot)$ с операцией конкатенации образует *моноид*. Если $a = xby$, где $a, x, b, y \in A^*$, то слово b называется *подсловом* или *фактором* слова a.

Если же $a = (\alpha_1\alpha_2 \dots \alpha_n)$ и $a' = (\alpha_{i_1}\alpha_{i_2} \dots \alpha_{i_k})$, где $1 \le i_1 < i_2 < \dots < i_k \le n$, то a' — это *фрагмент* слова a. Ясно, что каждое подслово слова a является одновременно и фрагментом a. Таким образом, и подслово, и фрагмент являются частями исходного слова, которые несут об этом слове определённую информацию.

Если $a = (\alpha_1\alpha_2 \dots \alpha_n)$, то длина слова a по определению равна n. При этом используется обозначение

$$|a| = n.$$

Отображение

$$x \to |x|$$

является гомоморфизмом моноида $(A^*,\cdot)$ в полугруппу натуральных чисел Z. При этом выполнены очевидные соотношения.

1) $|\Lambda| = 0$,

2) $|xy| = |x| + |y|$,

3) $|x^n| = n|x|$.

Бинарные отношения «быть подсловом» и «быть фрагментом» превращают моноид $(A^*,\cdot)$ в частично-упорядоченное множество. Диаграммы Хассе этих частичных порядков существенно различаются.

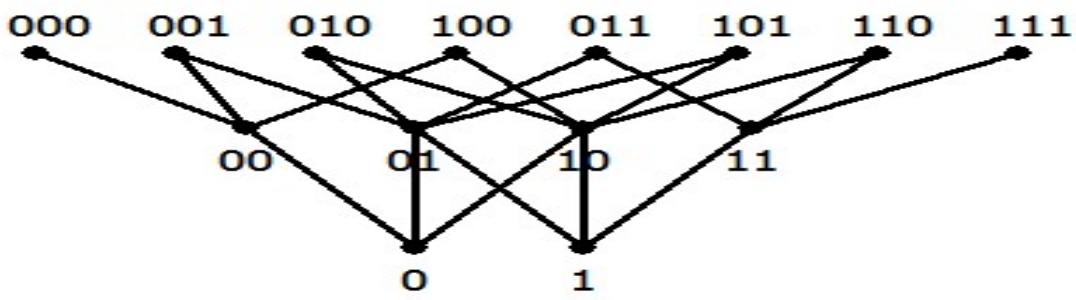
1) Частичный порядок — быть подсловом.

Мы будем писать

$$x \overset{s}{\le} y,$$

если слово x является подсловом слова y.

Пусть $A = \{0,1\}$ и $A^{\le n}$ — множество всех слов длины $\le n$ над бинарным алфавитом. Диаграмма Хассе Ч.У.М. $A^{\le 3}$ выглядит следующим образом (рис. 1):



Общая геометрическая характеристика диаграммы Хассе частичного порядка $\left(A^*, \overset{s}{\leq}\right)$ состоит в следующем.
a) Степень исхода вершины x равна четырём, если $x \neq \gamma^k$ и равна трём, если $x = \gamma^k$, где $\gamma \in A$.
b) Степень захода вершины x равна единице, если $x = \gamma^k$ и равна двум, если $x \neq \gamma^k$.

Таким образом, «вверх» из каждой вершины этой диаграммы Хассе идут три или четыре ребра, а «вниз» — одно или два ребра.

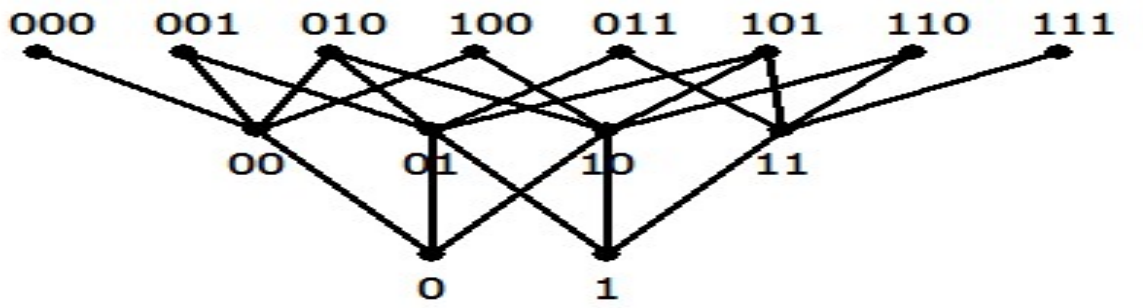
2) Частичный порядок — быть фрагментом.

Мы будем писать

$$x \leq y,$$

если слово x является фрагментом слова y.

Пусть $A = \{0,1\}$. Рассмотрим диаграмму Хассе следующего частичного порядка $(A^{\leq 3}, \leq)$.



Общая геометрическая характеристика диаграммы Хассе частичного порядка $(A^*, \leq)$ состоит в следующем.
a) Если x лежит в k-м ярусе диаграммы Хассе, то есть $|x| = k$, то степень исхода вершины x равна $|x| + 2$.

b) Если слово x имеет k серий, то есть $x = \gamma^{x_1}\overline{\gamma}^{x_2}\gamma^{x_3}\ldots\gamma^{\sigma x_k}$ где $x_i \geq 1$, то степень захода вершины x равна k.

Другое качественное различие диаграмм Хассе частичных порядков, определённых выше, заключается в следующем обстоятельстве.

Предложение. Любая антицепь в Ч.У.М. $(A^*, \leq)$ имеет конечное число элементов.

Рассмотрим теперь следующее множество слов W:

$$W = \{01^k0\}\ k = 1,2,\ldots$$

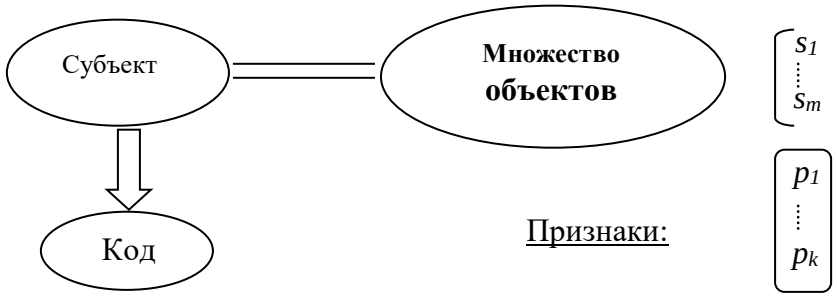
Предложение. Множество W является антицепью в частично-упорядоченном множестве $\left(A^*, \overset{s}{\leq}\right)$.

Таким образом, W — это бесконечное множество слов, каждое из которых не является подсловом никакого другого слова из W. Как было отмечено выше, такая ситуация невозможна в случае, когда частичный порядок задаётся отношением — «быть фрагментом».

3. Лекция 3. Признаковое кодирование. Тесты для таблиц.

3.1 Признаковое кодирование.

Метод признакового кодирования можно использовать в разных ситуация, но типичное его применение – кодирование «неформализованных» («нематематических») объектов.



Рассматривается множество объектов $s_1,\ldots,s_m$ (люди, болезни, месторождения и т.п.), которое нужно кодировать, т.е. каждому объекту нужно сопоставить слово в алфавите. При этом есть другое множество объектов, называемых признаками, которые мы умеем кодировать. Тогда кодом объекта при **признаковом кодировании** является вектор, компонентами которого являются коды (значения) признаков, применительно к данному объекту.

При признаковом кодировании объекта $x \in X$ признаком f называется отображение $f : x \to D_f$, где D_f — множество допустимых значений признака.

Если заданы признаки $f_1,\ldots,f_n$, то вектор $x=(f_1(x),\ldots,f_n(x))$ называется признаковым описанием объекта x . Если признаковые описания отождествляют с самими объектами, то множество $X = D_{f_1} \times \ldots \times D_{f_n}$ называют признаковым пространством.

В зависимости от вида множества признаки делятся на следующие типы:

1. бинарный признак: $D_f = (0,1)$;
2. номинальный признак: D_f - конечное множество;
3. порядковый признак: D_f - конечное упорядоченное множество;
4. количественный признак: D_f - множество действительных чисел.

Можно использовать характеристические вектора признаков, например,

$\alpha(s_i)=(\alpha_1, \ldots, \alpha_k)$ $\begin{cases}\alpha_i = 1, \text{ если } s_i \text{ обладает признаком } p_j \\ 0, \text{ иначе}\end{cases}$.

Пример. $s_1 \ldots s_n$ – люди.

Признаки:

1. Цвет волос
2. Рост
3. Вес
4. $p_4 \ldots p_n$ – размеры частей тела человека.

3.2 Тесты для таблиц.

В случае бинарных значений признаков или использовании характеристических векторов при признаковом кодировании объектом исследования является матрица векторов признаков.

$$T(A)=\begin{vmatrix} a_{11} & \ldots & \ldots \\ \ldots & \ldots & \ldots \\ \ldots & \ldots & a_{mn} \end{vmatrix}_{m\times n}$$

Определение. Тест - это множество столбцов в $T(A)$ такое, что все строки в подматрице, образованной этим множеством, попарно различны.

Выше мы уже говорили о стремлении строить неизбыточные коды. С этой задачей связана известная проблема дискретной математики – проблема поиска минимальных тестов таблицы.

Длина теста – количество столбцов в тесте.

Тест минимальной длины для A называется *минимальным* и его длина обозначается через $t(A)$. Тест T называется *тупиковым*, если никакое подмножество $T' \subseteq T$ тестом не является.

Когда таблица $T(A)$ – это признаковая таблица множества объектов (строки таблицы – вектора признаков объектов), то длина теста указывает на количество признаков, которых достаточно для различения объектов. Эти признаки соответствуют столбцам теста. Но будет ли их достаточно, если количество объектов увеличиться? В связи с этим вопросом *тестовая тематика* возникает и при определении качества выбора множества признаков.

Конечно, как математический объект, тест – это скорее тема таких дисциплин как комбинаторика, дискретная математика, теория распознавания, но, так как он тесно связан с проблемой грамотного представления информации, то это объект и теории информации.

Для получения содержательных результатов в области построения минимальных тестов необходимо накладывать ограничение на вид и структуру векторов признаков. В качестве примеров приведем несколько утверждений, доказательство которых либо очевидно, либо является несложным. (Если не оговорено обратное – основание логарифма равно двум). Т(А) – матрица из нулей и единиц.

Примеры. (Свойства тестов.)

1) Если E — единичная матрица, то любые $(n-1)$-столбцов образуют тест.

2) Если A — любая матрица перестановок, то есть $A \in C_n$, то любые $(n-1)$-столбцов A образуют тест.

3) Пусть A — второй слой в B^4. Тогда

$$A = \begin{Vmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{Vmatrix}$$

и первые три столбца A образуют тест.

Утверждение. Если $A = \left|\left|\alpha_{ij}\right|\right|_{m\times n}$ — матрица и расстояние Хэмминга между любыми двумя строками A больше, чем r, то $t(A) \le n - r$.

Доказательство. Выберем любые $(n-r)$-столбцов матрицы A и рассмотрим подматрицу A', порождённую этими столбцами. В этой подматрице все строки разные, то есть если $a' = b'$, то $\rho(a,b) \le r$, что противоречит исходному предположению.

Утверждение. Пусть $T(A)$ — матрица, полученная из A перестановкой некоторых строк или столбцов. Тогда $t(A) = t\big(T(A)\big)$.

Утверждение. Справедливо соотношение: $t(A) = t\big(\overline{A}\big)$, где $\overline{A}$ — логическое отрицание матрицы A, то есть $\overline{A} = \left|\left|\overline{\alpha_{ij}}\right|\right|$.

Утверждение. Если q — однородный (состоящий из одинаковых элементов) столбец матрицы A, то $t(A) = t(A - q)$.

Утверждение. Если $A = \left|\left|\alpha_{ij}\right|\right|_{m\times n}$, то $t(A) \ge log_2 m$.

Доказательство. Если столбцы $I_1, I_2, \dots, I_k$ — образуют тест, то все строки матрицы $A' = \left|\left|I_1, I_2, \dots, I_k\right|\right|$ — различны. Так как длина каждой такой строки равна k и общее число строк длины k есть 2^k, то $2^k \ge m$.

Что и требовалось доказать.

Заметим, что данное утверждение тесно связано с приведенным ниже понятием *энтропия по Хартли*. Оно означает, что если объектам даются битовые идентификаторы, то число бит в таком идентификаторе для различения m объектов не может быть меньше *log* m.

Лемма. Если строки матрицы $A = \left\|\alpha_{ij}\right\|$ образуют группу по сложению, то любой тупиковый тест является минимальным и $t(A) = log_2(m)$.

Доказательство. Пусть A — матрица с указанным свойством, где $m = 2^k$ и $I_1, I_2, \dots, I_k$ — произвольное подмножество из k-линейно независимых столбцов. Такое множество существует, поскольку ранг матрицы A равен k. Рассмотрим подматрицу A', состоящую из столбцов $I_1, I_2, \dots, I_k$. Ранг этой подматрицы равен k и потому все строки этой подматрицы различны, так как в A' есть k-линейно независимых строк таких, что все остальные строки — их линейная оболочка. Отсюда следует, что столбцы $I_1, I_2, \dots, I_k$ образуют тест A. Из свойства 2 (см. выше) следует, что $t(A) \ge k$ и потому построенный тест оптимален, и всё доказано.

Лемма доказана.

Следствие. Число тупиковых тестов матрицы A равно числу базисов n-мерного пространства B^n и выражается следующей величиной:

$$\frac{\prod_{i=0}^{k}(2^k - 2^i)}{k!}$$

Основные проблемы в теории тестов состоят в следующем.

1. Для заданной матрицы A построить тест минимальной длины.
2. Найти число тупиковых тестов произвольной матрицы A.

3.3 Тесты и матрица сравнений.

Определение. Множество T-столбцов матрицы называется *столбцовым покрытием* для матрицы A, если в подматрице A', образованной столбцами из множества T, нет нулевых строк. Длиной столбцового покрытия T называется мощность множества T, то есть $|T|$.

Определение. Глубиной матрицы A называется следующая величина:

$$\xi(A) = \min_{T} |T|. \quad (3.1)$$

Минимум в (3.1) берётся по множеству всех столбцовых покрытий матрицы A.

Отметим следующее обстоятельство. Если $A = \left\|\alpha_{ij}\right\|$ — произвольная булева матрица, то под матрицей A^2 мы будем понимать матрицу, строками которой являются попарные суммы по *mod 2* всех пар строк матрицы A. Эта матрица A^2 называется *матрицей сравнений*. Смысл введённой операции состоит в следующем.

Утверждение. Пусть m≥2, тогда справедливо соотношение: $t(A) = \xi(A^2)$,

где, как было введено выше, $\xi(B)$ — глубина матрицы B.

Доказательство. Индукция по m. Для случая m = 2

$$\begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \beta_1 & \beta_2 & \dots & \beta_n \end{pmatrix}$$

«Различимость» строк полностью определяется множеством единиц в сумме строк этой матрицы. Другими словами, «различимость» — это единицы в сумме, а ноль — это совпадение. Если эти две строки одинаковы, то в A нет никаких тестов, а в A^2 –столбцовых покрытий, т.е. $t(A) = \xi(A^2) = 0$.

Пусть утверждение справедливо для m строк и $t(A)=k$, т.е. подматрица, образованная столбцами $i_1,\dots,i_k$, в A образует минимальный тест. Тогда эти же столбцы образуют в A^2 столцовое покрытие. В образованной ими подматрице не может быть нулевых строк, так как это бы означало, что в аналогичной подматрице матрицы A есть одинаковые строки. По этой же причине из минимальности длины теста следует минимальность мощности столбцового покрытия.

Добавим $m+1$ –ю строку. Если по-прежнему подматрица, образованная столбцами $i_1,\dots,i_k$, в новой матрице A образует тест, то вновь эти же столбцы образуют в новой матрице A^2 столцовое покрытие. Заметим, что при добавлении строки длина теста не может уменьшиться. Поэтому и тест, и покрытие будут минимальными. Если же указанные столбцы уже теста не образуют, то они не образуют в A^2 и столбцового покрытия. Но тогда мы ищем в A минимальный тест, а рассуждениями, аналогичными приведенным выше, можно показать, что столбцы этого теста образуют в A^2минимальное столбцовое покрытие.

Утверждение доказано.

3.4 Длина минимального теста для «почти всех» матриц.

Следующее вспомогательное утверждение, называемое неравенством Бернулли (неравенств с таким названием несколько!), бывает полезным в различного рода комбинаторных конструкциях.

Лемма (Бернулли). Пусть даны числа $\alpha_i, i = 1, \dots, N, \alpha_i \geq 0, \alpha_i < 1$. Справедливо неравенство

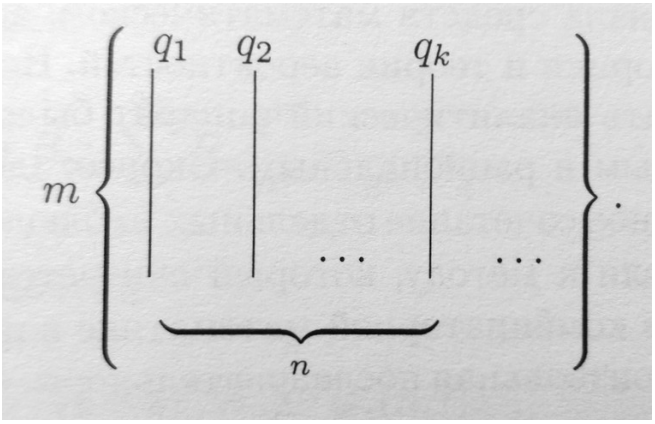
$$\prod_{i=1}^{N}(1-\alpha_i) \geq 1 - \sum_{i=1}^{N}\alpha_i.$$

Доказывается по индукции совсем просто.

В одном из приведенных выше примеров длина минимального теста равнялась m-1, а простая нижняя оценка это величины $t(A) \geq log_2 m$. А если взять матрицу случайно, то что может быть? Следующая теорема дает ответ на этот вопрос.

Теорема. Для почти всех $(m \times n)$-матриц длина максимального теста не превосходит $(2+\epsilon)log_2 m$.

Доказательство. Выделим некоторый набор столбцов $T = \{q_1 q_2 \dots q_k\}$ и найдём число матриц A, у которых эта совокупность является тестом.



Искомое число матриц равно следующей величине:

$$l_k(m,n) = 2^k(2^k-1)\dots(2^k-m+1)2^{mn-mk} =$$
$$= 2^{mk}\left(1-\frac{1}{2^k}\right)\left(1-\frac{2}{2^k}\right)\dots\left(1-\frac{m-1}{2^k}\right)2^{mn-mk} =$$
$$= \prod_{i=1}^{m-1}\left(1-\frac{i}{2^k}\right)2^{mn} > \left(1-\sum_{i=1}^{m-1}\frac{i}{2^k}\right)2^{mn}.$$

Тут использовалось неравенство Бернулли:

$$\prod_{i=1}^{N}(1-\alpha_i) \geq 1 - \sum_{i=1}^{N}\alpha_i.$$

Здесь мы учли тот факт, что столбцы $T = \{q_1 q_2 \dots q_k\}$ образуют тест, поэтому справедливо неравенство $k \geq log_2 m$. Тогда, если $\alpha_i = \frac{i}{2^k}, i = 1,\dots,m-1$, то выполняются условия применения этого неравенства Бернулли: $\alpha_i, i = 1, \dots, m-1, \alpha_i \geq 0, \alpha_i < 1$.

Далее имеем

$$\sum_{i=1}^{m-1} \frac{i}{2^k} < \frac{1}{2^k}\frac{m^2}{2} = \frac{m^2}{2^{k+1}},$$

и при $k = (2+\epsilon)log_2 m$ имеем

$$\frac{m^2}{2^k} = \frac{m^2}{m^{2+\epsilon}} \to 0 \text{ при } m \to \infty.$$

Таким образом,

$$\frac{l_k(m,n)}{2^{mn}} \to 1$$

при

$$k \geq (2+\epsilon)log_2 m$$

Теорема доказана.

3.5 Упражнения.

Пусть $T(A) = \begin{vmatrix} a_{11} & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & a_{mn} \end{vmatrix}_{m \times n}$ - матрица из нулей и единиц.

1. Пусть строки матрицы T(A) устроены так, что вместе с любой парой строк в матрице есть и строка, равная сумме этой пары по модулю два. Доказать, что в такой матрице любой тупиковый тест будет минимальным и

$$t(A)=\log m.$$

2. Число тупиковых тестов таблицы не превосходит $C_n^{[n/2]}$.

3. Пусть строками матрицы являются все наборы с четным числом единиц. Найти t(A) и число тупиковых тестов.

4. Пусть строками матрицы являются все наборы с фиксированным числом единиц. Найти t(A) и число тупиковых тестов.

4. Лекция 4. Кодирование слов. Комбинаторика слов и фрагментов.

4.1. Кодирование слов.

Как уже выше говорилось, любая информация может быть представлена словом в конечном алфавите. Есть объект *O*, субъект *S*, а перед субъектом стоит задача *Z*, связанная с данным объектом. Субъект решает эту задачу с помощью алгоритма *A(Z)*, входом которого является информация об объекте в виде слова *I(O)*. Как мы уже выше видели на примерах чисел, графов и признаковых векторов информацию эту об объекте можно представлять по-разному: разными словами в разных алфавитах. Предполагается, конечно, что любое такое представление *дешифруемо*, *неизбыточно* и *разумно*.

Но предположим, что имеется критерий предпочтения одних представлений другим, например, минимизация длины этого самого слова-представления. Возникает задача преобразования одного слова *I(O)* в другое *J(O)*.

В этом случае рассматривается следующая модель. Есть два алфавита

$$A = (a_1, \dots, a_n) \text{ и } B = (b_1, \dots, b_q).$$

Кодирование φ это отображение слов α в алфавите A в слова $\beta=\varphi(\alpha)$ в алфавите B.

Кодирование будет дешифруемым, если по слову β однозначно восстанавливается α. Тогда отображение φ – биекция.

Например, у нас есть кодируемые объекты *x* - слова в алфавите *A*. Есть другой алфавит, например, бинарный. Нельзя ли найти минимальную длину K(x) кода объекта x, пусть даже в другом алфавите B?

Простейший случай. Есть два алфавита. Когда кодирование более «экономно»?

Пример. *Сериальное кодирование*.

Здесь $A = (0,1)$ *и* $B = (0,1,\dots,9,*)$. Символ * играет роль разделителя.

В данном примере битовые последовательности кодируются натуральными числами. Как это следует из свойств данного кодирование, оно дает эффект сжатия, когда битовые последовательности представляют собой чередование сравнительно больших блоков нулей и единиц.

Алгоритм кодирования. Пусть дана битовая последовательность α и $a \in (0,1)$ ее первый символ. Серией назовем часть последовательности, состоящую из одинаковых символов. Тогда любую битовую последовательность можно представить, как чередование серий из нулей и единиц. Пусть в последовательности α всего S серий, длина первой - x_1, длина второй - x_2,…, длина S–й - x_S. Тогда кодом α будет слово $\beta=\varphi(\alpha)=a^* x_1^* x_2^* \dots^* x_S$.

Алгоритм декодирования очевиден.

Ниже мы более подробно проанализируем свойства сериального кодирования.

4.2. Слова и языки

Существует много способов получать из одних слов другие слова. Так возникают специальные классы слов, языки и другие алгебраические конструкции. На эту тему у вас был обширный и глубокий курс лекций и семинаров. Приведем лишь несколько примеров.

Пример. Если $A = \{0,1\}$ и μ —морфизм,

$$\mu(0) = 01, \mu(1) = 10,$$

то итерации морфизма μ порождают слова по стандартным правилам:

$\mu^2(0) = \mu\big(\mu(0)\big) = \mu(01) = \mu(0)\mu(1) = 0110,$

$\mu^2(1) = 1001,$

$\mu^3(0) = \mu\big(\mu^2(0)\big) = \mu(0110) = 01101001,$

$\mu^3(1) = 10010110.$

Аналогичным образом могут быть найдены все итерации морфизма μ. Бесконечные слова

$t = \mu^{\omega}(a)$ и $\overline{t} = \mu^{\omega}(b)$ называются словами Туэ — Морзе.

Другим способом построения языков является использование частей одного и того же слова.

Пример. Пусть $B = \{0,1\}$ и $a \in B^n$. Рассмотрим последовательность всех фрагментов длины k слова a, выписанных в лексикографическом (по способу построения) порядке:

$$v_1, v_2, \dots, v_{\binom{n}{k}}.$$

Положим по определению

$$(a)_k = v_1 v_2 \dots v_{\binom{n}{k}}.$$

Таким образом, $(a)_k$ — это конкатенация всех фрагментов длины k у слова a, выписанных в лексикографическом порядке. Длина слова $(a)_k$ равна $k\binom{n}{k}$.

Применяя к слову $(a)_k$ то же самое преобразование, мы получим слово $(a_1)_k = ((a)_k)_k$. Так формируется язык $L_k(a) = \{a, (a)_k, ((a)_k)_k \dots\}$.

Отметим далее, что отображение $T_k(a)$:

$$B^n \to B^{k\binom{n}{k}}, (4.1)$$

заданное формулой (4.1), является линейным, и если фрагменту v_r сопоставить матрицу A_r такую, что

$$aA_r = v_r r = 1{,}2$$

то матрица A линейного преобразования является конкатенацией матриц A_r, то есть

$$A = \left\| A_1 A_2 \dots A_{\binom{n}{k}} \right\|.$$

Пример

1) Пусть $n = 3$, $k = 2$ и фрагменты длины два упорядочены лексикографически: (12), (13), (23). Фрагмент (12) реализуется матрицей A_1, (13) — A_2, (23) — A_3, где

$$A_1 = \left\| \begin{matrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{matrix} \right\|, A_2 = \left\| \begin{matrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{matrix} \right\|, A_3 = \left\| \begin{matrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{matrix} \right\|.$$

Поэтому матрица A линейного преобразования

$$B^3 \to B^6$$

выглядит так:

$$A = \left\| \begin{matrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{matrix} \right\|.$$

Ясно, что

$$(a_1 a_2 a_3)A = (a_1 a_2 a_1 a_3 a_2 a_3).$$

«Геометрические» свойства преобразования $T_k(a)$ описываются в следующих утверждениях.

Предложение. $\|(a)_k\| = \|a\| \binom{n-1}{k-1}$.

Доказательство. Пусть $||a|| = m$. Тогда число фрагментов длины k у слова a, имеющих вес, равный r, равно $\binom{m}{r}\binom{n-m}{k-r}$. Поэтому

$$\|(a)_k\| = \sum_{r=1}^{m} r\binom{m}{r}\binom{n-m}{k-r} = m\sum_{r=1}^{m}\binom{m-1}{r-1}\binom{n-m}{k-r} = m\binom{n-1}{k-1}.$$

Предложение. $\frac{\|(a)_k\|}{|(a)_k|} = \frac{\|a\|}{|a|} k = 1,2, \dots, |a|$.

Предложение. $\rho((a)_k, (b)_k) = \binom{n-1}{k-1} \rho(a, b)$..

Выбор фрагмента в качестве «части» слова при построении языка $L_k(a)$ является реализацией лишь одного из возможных способов построения языка, исходя из фиксированного слова и его частей.

Выбрав вместо фрагмента «подслово» фиксированной длины, можно получить другие языки с теми или иными структурными особенностями.

Разумеется, существует много разных формальных языков и много разных конструкций, позволяющих строить и анализировать эти языки.

4.3. Слова и фрагменты

Если $x, a \in A^*$, то под биномиальным коэффициентом $\binom{x}{a}$ понимается число вхождений слова a в качестве фрагмента в слово x. Так как

$$\binom{1^n}{1^m} = \binom{n}{m},$$

то функция $\binom{x}{a}$ является, в определённом смысле, обобщением биномиального коэффициента, что и послужило основанием для использования классического обозначения $\binom{x}{a}$.

Для произвольного слова $x \in A^n$ множество чисел вида $\binom{x}{a}$, где $|a| \leq |x|$, называются параметрами Уитни слова x. Эти параметры играют значительную роль как в комбинаторике слов, так и в различных приложениях.

Пример

2) $\binom{10100}{010} = 2, \binom{011101}{01} = 5,$

$\binom{1^p 0^q}{10} = pq, \binom{1^m 01^n}{1^p} = \binom{m+n}{p}.$

Следующие свойства являются прямыми следствиями определений:

1) $\binom{x}{a} = \binom{\overline{x}}{\overline{a}}$;

2) $\binom{x}{1} = ||x||, \binom{x}{0} = |x| - ||x||$;

3) $\binom{x\alpha}{y\beta} = \binom{x}{y\beta} + \delta_{\alpha,\beta} \binom{x}{y}.$

Здесь $\alpha, \beta \in A = \{0,1\}$, $\delta_{\alpha,\beta}$ — символ Кронекера.
Далее в этом разделе всюду рассматривается только двоичный алфавит.

Обсудим теперь следующую проблему: как формально описать вычисление биномиального коэффициента

$$\binom{x}{a} = \binom{x_1 x_2 \dots x_n}{\alpha_1 \alpha_2 \dots \alpha_m}.$$

Пусть $a = \alpha_1 \alpha_2 \dots \alpha_m$ и

$$a_k = 1 - \alpha_k, b_k = 2\alpha_k - 1, k = \overline{1, m}.$$

Лемма. Справедлива формула

$$\binom{x_1 \dots x_n}{\alpha_1 \dots \alpha_m} = \sum_{1 \le i_1 < \dots < i_m \le n} (b_1 x_{i_1} + a_1) \times (b_2 x_{i_2} + a_2) \dots (b_m x_{i_m} + a_m). \quad (4.2)$$

Формула (4.2) свидетельствует о том, что биномиальный коэффициент может быть выражен в виде полинома от переменных $\{x_1 x_2 \dots x_m\}$ с целыми коэффициентами.

Следствие. Справедливы соотношения

$$\binom{x}{10} = \sum_{i<j} x_i (1 - x_j), \binom{x}{01} = \sum_{i<j} (1 - x_i) x_j,$$

$$\binom{x}{00} = \sum_{i<j} (1 - x_i)(1 - x_j), \binom{x}{11} = \sum_{i<j} x_i x_j.$$

Лемма дается без доказательства. Доказательство соотношений из данного следствия можно провести без использования леммы. Это предлагается вам сделать в качестве задач из задания. Сделав эти задачи, вы легко поймете смыл формулы (4.2).

Полином $\binom{x_1 \dots x_n}{\alpha_1 \dots \alpha_m}$ может быть представлен в виде суммы моногенных слагаемых, по которым восстанавливается весь полином. Возвращаясь к предыдущему примеру, мы находим эти полиномы в следующей форме:

$$\sum_{1 \le i < j \le n} x_i, \sum_{1 \le i < j \le n} x_j, \sum_{1 \le i < j \le n} x_i x_j,$$

или, что то же самое,

$$\sum_{i=1} \binom{n-i-1}{1} x_i, \sum_{j=1} \binom{i-1}{1} x_j,$$

$$\binom{||x||}{2} = \frac{(\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} x_i - 1)}{2}.$$

В общем виде ситуация выглядит следующим образом.

Лемма. Справдлива формула

$$\sum_{1 \le i_1 < i_2 < \dots < i_k \le n} x_{j_1} x_{j_2} \dots x_{j_s} = \sum_{1 \le j_1 < j_2 < \dots < j_s \le n} \binom{i_1 - 1}{j_1 - 1} \binom{i_2 - i_1 - 1}{j_2 - j_1 - 1} \dots \binom{n - i_s}{k - j_s} x_{j_1} \dots x_{j_s}.$$

Здесь $(j_1 j_2 \dots j_s)$ — подпоследовательность последовательности $(i_1 i_2 \dots i_k)$.

Доказательство. Так как $(j_1 \dots j_s)$ — подпоследовательность $(i_1 i_2 \dots i_k)$, то j_1 может быть выбран $\binom{i_1 - 1}{j_1 - 1}$-способами, j_2 — $\binom{i_2 - i_1 - 1}{j_2 - j_1 - 1}$-способами, …, j_s — $\binom{n - i_s}{k - j_s}$-способами. Что и требовалось доказать.

Утверждение. Если $x = (\alpha_1 \alpha_2 \dots \alpha_n) \in A^n = \{0,1\}^n, ||x|| = k, y_1 \dots y_k$ — номера единиц в слове x и $e_r = 1^{n-r} 0 1^{k-r+1}$, то

$$y_s = y_{s-1} + \binom{x}{e_s} + 1 s = 1,2, \dots, k.$$

Доказательство этого утверждения будет приведено в следующих лекциях. Смысл утверждения состоит в том, что номера единичных букв в слове x могут быть выражены в терминах числа фрагментов определённого вида и длины. В частности, если $||x|| = k$, то слово x , как будет показано далее, может быть восстановлено по фрагментам длины k.

4.4. Кодирование слов. О минимальной длине кода слова.

Пусть $\Sigma = \{a_1 a_2 \dots a_m\}$ и $x \in \Sigma^*$. Под кодом слова x мы будем понимать любое другое слово y, по которому слово x можно восстановить однозначно. Например, если x — бинарное слово длины n и $y_1 y_2 \dots y_m$ — номера единиц слова x, то по слову $y = (y_1 y_2 \dots y_m)$ слово x можно восстановить однозначно. Следующее определение связывает

слова и слова.

Определение. (А.Н. Колмогорова.) Пусть $F(x)$ — произвольная частично-рекурсивная функция (Машина Тьюринга). Тогда «сложность» слова a по $F(x)$ есть следующая величина:

$$K_F(a) = \begin{cases} \min l(p): & F(p) = a, \\ \infty, & \text{если } \forall p\colon F(p) \neq a. \end{cases} \qquad (4.3)$$

Любое слово p, удовлетворяющее условию (4.3), называется кодом или программой для слова a.

Таким образом, согласно (4.3), для восстановления «сложных» слов нужны длинные программы, а «простые» слова имеют короткие коды. Однако это определение включает в себя некоторый произвол, связанный с функцией $F(x)$. Этот произвол состоит в том, что мы имеем дело с определённой машиной Тьюринга(Т-М). Таким образом, если $F(x) = y$, то x является кодом слова y относительно Т-М, обозначаемой через P. Длина самого короткого кода обозначается через $K_p(y)$. Существование универсальной машины Тьюринга позволяет, в определённой степени, получить меру сложности, инвариантную относительно P.

Теорема. Существует такая машина Тьюринга S, что

$$K_s(y) \leq K_p(y) + C,$$

где константа C зависит лишь от МТ P.

Казалось бы, что мера (4.3) позволяет адекватно и конструктивно оценивать алгоритмическую сложность кодирования произвольного слова. Однако это не совсем так, в силу следующего утверждения.

Теорема. Функция $K(x)$, измеряющая сложность слова x относительно какой-нибудь универсальной машины Тьюринга, является алгоритмически неразрешимой.

Доказательство. Рассмотрим бинарный алфавит $B = \{0,1\}$ и упорядочим все слова из B^* сначала по длине, а потом по возрастанию (лексикографически).

$$0,1,00,01,10,11,000,001,010,011,100,101,110,111, \dots (*)$$

Если существует Т-М P, которая вычисляет $K(x)$, то существует и Т-М P_1, которая «переводит» любое натуральное число m в первое по порядку (*) слово x такое, что $K(x) \geq m$. Но тогда по числу m можно восстановить слово x, то есть m является кодом слова x. Так как длина m есть $log_2 m$, то справедливо неравенство $K(x) < log_2 m$. Противоречие.

Замечание. Нетрудно понять, что существует лишь конечное число слов, сложность которых равна m. Действительно, если не так, то существуют слова, имеющие одинаковые коды, что противоречит условию однозначности кодирования.

Пусть $A = \{a_1 \dots a_n\}$ — конечное множество. Под кодированием A понимается любое инъективное отображение A в некоторое множество бинарных слов. Другими словами, мы рассматриваем отображения вида f:

$$A \xrightarrow{f} B^*,$$

где $B = \{0,1\}$ и $f(x) \neq f(y)$ при $x \neq y$.

Длина кода $f(x)$ слова $x \in A$ — это, по определению, длина слова $f(x)$, то есть $|f(x)|$. Для того чтобы различать элементы множества A, коды этих элементов должны иметь

определённые ограничения на длину. Точный смысл этого замечания состоит в следующем.

Лемма. Справедливы соотношения

$$1.\ \max_{x\in A}|f(x)| \geq log_2|A|,$$

$$2.\ |x: f(x) \leq log|A| - t| < 2^{-(t-1)} \cdot |A|.$$

Доказательство. Так как число двоичных слов длины $\leq m$ равно

$$1 + 2 + \cdots + 2^m = 2^{m+1} - 1,$$

то если $|f(x)| < r$, тогда для кодирования A имеется не более $1 + 2 + \cdots + 2^{r-1} = 2^r - 1$ двоичных слов. Так как $|A| = n$, то должно выполняться неравенство: $2^r - 1 \geq n$, что соответствует условию 1).

С другой стороны, словами длины $(log_2 n - r)$ можно закодировать не более чем $2^{log_2 n - r} = \frac{|A|}{2^r}$ элементов множества A.

Лемма доказана.

4.5. Упражнения.

1. Найдите:

а) число бинарных слов длины *n*, имеющих ровно *k* серий;

б) число бинарных слов длины *n*, имеющих ровно *k* единичных серий;

в) число слов длины *n* над алфавитом $\{a_1,\ldots,a_m\}$, имеющих ровно *k* серий.

2. Пусть $a=(a_1\ a_2\ \ldots,a_n)$ – бинарное слово. Показать, что:

а) число фрагментов вида (11) в слове *a* равно $\binom{\|a\|}{2}$, где $\|a\|$ – число единиц в слове *a;*

б) число фрагментов вида (01) в слове *a* равно $\sum_{i=1}^{n} ia_i - \binom{\|a\|+1}{2}$.

3. Пусть $A=\{1,2,\ldots, m\}$. Слово $a=(a_1,\ldots,a_n)$ над алфавитом *A* называется монотонным, если $a_1 \leq a_2 \leq a_3 \leq \ldots \leq a_n$. Найти число монотонных слов длины *n* над алфавитом *A*.

4. Пусть $a=(a_1\ ,a_2\ \ldots,\ a_n)$ – бинарное слово. Найти:

а) общее число фрагментов слова a;

б) число различных фрагментов слова $a=(11\underbrace{\ldots}_{k}10\underbrace{\ldots}_{n-k}0)$.

5. Покажите, что

а) Число слов длины n, содержащих в качестве фрагмента данное слово x длины k, равно

$$\sum_{i=k}^{n}\binom{n}{i}(m-1)^{n-i}.$$

b) Общее число подслов длины k в слове $x \in A^n$ равно $n - k + 1$.

c) Общее число фрагментов длины k в слове $x \in A^n$ равно $\binom{n}{k}$.

5. Лекция 5. Примеры кодирований: сериальное кодирование, кодирование натуральных чисел, побуквенное кодирование.

5.1. Сериальное кодирование.

Определение сериального кода было дано в предыдущей лекции. Рассмотрим его свойства.

Определение. Длиной сериального кода назовем число $l(\alpha) = \sum_{i=1}^{S} [\log x_i]^*$, где

$[log\ x]^*=1$ при $x=1$ и $[log\ x]^*= [log\ x]$ при $x>1$.

Определение. Средней длиной кода для n-последовательностей назовем число

$$\tilde{l} = \frac{1}{2^n} \sum_{\alpha \in B^n} l(\alpha),$$

Где сумма берется по всем двоичным последовательностям длины n.

Свойства такого кодирования иллюстрируются следующими двумя утверждениями.

Утверждение. Если все $x_i=1$, то $l(\alpha)=|\alpha|$.

Утверждение. Если все $x_i>1$, то $l(\alpha) \le S \log \frac{n}{S}$.

Доказательство.

$$l(\alpha) = \sum_{i=1}^{S} [log x_i] \le \log(x_1 x_2 \ldots\ldots x_S) \le S \log \frac{x_1 + x_2 + \ldots\ldots + x_S}{S} = S \log \frac{n}{S}.$$

Утверждение. Средняя длина слова с S сериями в сериальном кодировании

$$\tilde{l}_{S,n} = \frac{S(S-1)}{n-S+1} + \frac{1}{C_{n-1}^{S-1}} \sum_{r=2}^{S} \binom{n-r-1}{S-2} [\ln r].$$

Утверждение. Справедлива асимптотическая оценка

$$\tilde{l}_{S,n} = \frac{S^2}{n-S} + \frac{S(n-S)}{n^2} \log(n-S).$$

Из этих утверждений следует.

Если $S = const$, *то длина* $l(\alpha) = O(\log n)$

Если $S = kn\ (k<1) \Rightarrow\ l(\alpha) = O(S) = O(n)$

Таким образом, сериальное кодирование при малых количествах серий и большой длине серий приводит к тому, что при таком кодировании возникает эффект сжатия.

Если серии короткие и их много, например: $s = k_n$ (при $k < 1$) и $l(\alpha) = O(\mathrm{s}) = O(\mathrm{n})$, то эффекта сжатия не возникает.

Пусть $t_n(m,k)$ — число слов из B^n веса m, имеющих ровно k серий.

Лемма. Справдлива формула

$$t_n(m,k) = \begin{cases} \binom{m-1}{r}\binom{n-m-1}{r-1} + \binom{m-1}{r-1}\binom{n-m-1}{r} \text{ при } r \neq 0 \text{ и } k = 2r+1; \\ 2\binom{m-1}{r}\binom{n-m-1}{r} \text{ при } r \neq 0 \text{ и } k = 2r; \\ 1 \text{ при } k = 1. \end{cases} \quad (5.2)$$

Доказательство. Если слово $x \in B^n$ имеет k серий, то

$$x = \gamma^{t_1}\gamma^{t_2} \dots \gamma^{t_k},$$

где $\gamma \in B = \{0,1\}$ и $\sum_{i=1}^{k} t_i = n$ и $t_i \geq 1$ для $i = \overline{1,k}$.

При этом если $\gamma = 1$, то

$$||x|| = t_1 + t_3 + \cdots,$$

и если $\gamma = 0$, то

$$||x|| = t_2 + t_4 + \cdots$$

Далее, если $\gamma = 1$ и $k \equiv 1\ (mod 2)$, то $t_n(m,k)$ равно числу решений системы уравнений:

$$\begin{cases} t_1 + t_3 + \cdots + t_k = m, \\ t_2 + t_4 + \cdots + t_{k-1} = n - m. \end{cases} \quad (5.3)$$

Если $k = 2r+1$, то (5.3) переходит в следующую форму:

$$\begin{cases} t_1 + t_3 + \cdots + t_{2r+1} = m, \\ t_2 + t_4 + \cdots + t_{2r} = n - m. \end{cases} \quad (5.4)$$

Число неизвестных в первом уравнении равно $(r+1)$, а во втором — r.

Так как множества неизвестных не пересекаются, то число решений системы (5.4) равно $\binom{m-1}{r}\binom{n-m-1}{r-1}$. Если $\gamma = 0$ и $k \equiv 1\ (mod 2)$, то $t_n(m,k)$ равно числу решений системы уравнений

$$\begin{cases} t_2 + t_4 + \cdots + t_{k-1} = m, \\ t_1 + t_3 + \cdots + t_k = n - m \end{cases}$$

или

$$\begin{cases} t_1 + t_3 + \cdots + t_{2r+1} = n - m, \\ t_2 + t_4 + \cdots + t_{2r} = m. \end{cases} \quad (5.5)$$

Таким образом, число решений системы (4.5) равно $\binom{n-m-1}{r}\binom{m-1}{r-1}$. Поэтому в случае $k \equiv 1\ (mod 2)$, то есть $k = 2r+1$, число $t_n(m,k)$ задаётся следующей формулой:

$$\binom{m-1}{r}\binom{n-m-1}{r-1} + \binom{m-1}{r-1}\binom{n-m-1}{r}$$

при $r \neq 0$. Если же $r = 0$, то есть $k = 1$, то $t_n(n,1) = 1$.

Если же $k \equiv 0\ (mod\ 2)$, то есть $k = 2r$, то, повторяя предыдущие рассуждения, получаем следующее выражение для $t_n(m,k)$:

$$2\binom{m-1}{r}\binom{n-m-1}{r}.$$

Окончательно получаем формулу (5.2).

Лемма доказана.

5.2. Коды натурального ряда.

Здесь мы будем рассматривать алфавиты $A = (0,1,\ldots,9)$ и $B = (0,1)$.

Задача кодирования натуральных чисел битовыми последовательностями актуальна, так как часто необходимо хранить и передавать информацию в бинарном виде. Особенно остро она встала в 50-е годы прошлого века в связи с распространением компьютеров.

В данной модели предполагается, что нужно работать (хранить, передавать) с битовой информацией, изначально представимой, например, в десятичных числах. При этом набор чисел представляется битовой последовательностью без разделителей.

Простой переход к двоичной записи не обеспечивает префиксности, а, следовательно, и декодируемости этой битовой последовательности.

Приведем примеры кодов, которые это обеспечивают.

Пусть $n > 1$, двоичную запись числа n обозначим $BIN(n) = \alpha_1 \ldots \alpha_n$, $\alpha_i \in \{0, 1\}$. (Например, BIN(30) = 11110, BIN(75) = 1001011). Заметим, что $k = |BIN(n)| = [\log n] + 1$. Пусть $B(n)$ – двоичная запись n без первого символа. (Например, B(30) = 1110, B(75) = 001011). Тогда $|B(n)| = [\log n]$.

Так как любая двоичная запись натурального числа начинается с единицы, то префиксность можно обеспечить с помощью тривиального кода

$$TK(n) = 0\ldots0BIN(n),$$

Где число нулей префиксе равно $[\log n] + 1$. Длина такого кода $2[\log n] + 2$. Следующие два примера дают более экономное кодирование.

5.2.1. Код Элайеса.

Код Элайеса неотрицательного целого числа n обозначим через $El(n)$. Положим $El(0) = 10$, $El(1) = 11$.

Пусть $n > 1$. В этом случае длина слова BIN(n) больше единицы. Код Элайеса состоит из трех фрагментов. Вначале идет некоторое количество нулей, подсчитав которые, мы найдем число символов в двоичной записи |BIN(n)|. Для экономии, учитывая предыдущее замечание можно взять количество нулей на единицу меньшее числа символов в двоичной записи |BIN(n)|. (Так как любая двоичная запись начинается с единицы, то мы всегда знаем, где заканчивается наша цепочка нулей). Теперь мы это количество символов записи |BIN(n)| отсчитываем после последнего нуля, а затем полученный фрагмент из нулей и единиц переводим в десятичную запись и получаем длину закодированного слова. Зная эту длину, мы отсчитываем нужное нам число символов, а полученный фрагмент и будет двоичной записью закодированного натурального числа. Опять же для экономии первую единицу данного фрагмента можно в коде опустить.

Тогда $El(n) = 00\ldots0\ BIN([\log n]+1)\ B(n)$, где 00…0 – маркер, необходимый для того, чтобы знать, сколько последующих за 0 символов может задавать длину записи числа. Поэтому в данном коде количество нулей перед $BIN([\log n]+1)$ равно $|BIN([\log n]+1)|-1$.

Примеры (пробелы стоят только для иллюстрации):

$El(5) = \underline{0}\ 11\ \underline{01.}$

$El(75)$ = 00 111 001011.

Утверждение. Длина $|El(n)|$ не превосходит $\log n + 2 \log (\log n) + 3$.

5.2.2. Код Левенштейна

В коде Элайеса мы сэкономили по сравнению с тривиальным кодом за счет уменьшения префикса. В коде Левенштейна эта идея доведена до определенного истощения: от двоичной записи числа мы сначала переходим к двоичной записи длины BIN(n) (это было и в коде Элайеса), но затем мы переходим к длине длины, длине длины длины и т.д.

Введем для удобства формальной записи этой идеи некоторые обозначения. Пусть $\lambda_0(n) = [\log n]$. А далее по аналогии до $\lambda^k{}_0(n) = \lambda_0 (\lambda^{k-1}{}_0(n)) = [\log \ldots [\log n]]$.

Для любого n существует S такое, что: $\lambda^S{}_0(n) = 0$, $\lambda^{S-1}{}_0(n) = 1$.

Положим $Lev(0)=0$, $Lev(1) =10$. Пусть $n > 1$. Тогда для такого сила вышеупомянутый параметр S>1. Если в префиксе кода мы ставим S подряд идущих единиц, а затем ноль (чтобы показать, где эта цепочка единиц заканчивается), то это не может быть ни кодом 0, ни кодом 1. А так как $\lambda^S{}_0(n) = 0$, $\lambda^{S-1}{}_0(n) = 1$, то эти соотношения никакой информации для кодирования не содержат, и в код надо включать информацию о длинах, начиная с $B(\lambda^{S-2}{}_0(n))$. Отсюда и следует формула для кода Левенштейна.

$$Lev(n) = 11\ldots10\ B(\lambda^{S-2}{}_0(n))\ldots B(\lambda_0(n))B(n),$$

где 11…10 – слово из S единиц и одного нуля.

Утверждение. Длина кода Левенштейна задается соотношением

$$|Lev(n)| = \log n + \log \log n + o(\log \log n).$$

Пример (пробелы только для иллюстрации):

$Lev(75)$ = 11110 0 11 001011

$\lambda_0(75) = 7$

$\lambda_0^{\,2}(75) = 2$

$\lambda_0^{\,3}(75) = 1$

$\lambda_0^{\,4}(75) = 0$

$S = 4$

Lev(5)=1110 0 01. Lev(62)=11110 0 01 11110.

5.3. Побуквенное кодирование.

Мы видели, что при словарном кодировании объектов найти код минимальной длины – это решить алгоритмически неразрешимую проблему. В такой ситуации уместна более слабая постановка задачи. Так как кодирование – это отображение одного множества слов в другое, то ослабление задачи заключается в наложении ограничений на вид такого отображения.

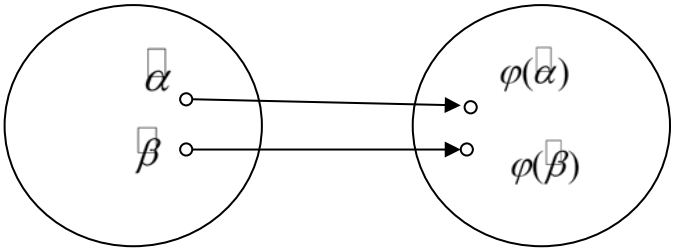
В предыдущей лекции мы рассмотрели два примера таких ограничений: сериальное кодирования и кодирование натурального ряда. Второй из этих примеров является частным случаем более общей ситуации: *префиксного* кодирования.

Если пренебречь разделителями, то в качестве двух алфавитов возьмем алфавиты

$$A = (a_1, \ldots, a_n) \text{ и } B = (b_1, \ldots, b_q).$$

Алгоритм побуквенного (алфавитного) кодирования. Каждой букве a_i алфавита А ставится в соответствие $B_i = \varphi(a_i)$ – слово в алфавите В длины l_i.

Алфавитное кодирование будет дешифруемым, если отображение φ таково, что для $\forall\, \alpha \neq \beta:\ \phi(\alpha) \neq \phi(\beta)$.



Заметим, что в приведенном определении термин *дешифруемость* используется в другом смысле, нежели выше при описании основных свойств кодирования (*дешифруемость – неизбыточность - разумность*). Поэтому в других учебниках используются иные термины, например, *разделимость*. Но не будем загромождать изложение терминами, тем более, что из контекста всегда ясно, о чем идет речь.

Обратим внимание на следующий важнейший факт, который, по сути дела, и оправдывает интерес к такому виду кодирования. Очевидно, что ограничение на вид отображения φ сразу влечет дешифруемость однобуквенных (в алфавите А) слов. Но из дешифруемости однобуквенных слов сразу следует дешифруемость слов (в алфавите А) любой длины.

Заметим также, что в случае дешифруемого кодирования все слова $B_i = \varphi(a_i)$ обязательно различны.

Оказывается, что *дешифруемость* накладывает ограничение на набор длин кодовых слов. Об этом говорит утверждение, известное как неравенство Крафта.

Теорема. Если алфавитное кодирование с длиной кодов $l_1, \ldots, l_n$ дешифруемо, то выполняется неравенство:

$$\sum_{i=1}^{n} q^{-l_i} \leq 1.$$

Доказательство. Пусть $A = (a_1, \ldots, a_n)$ *и* $B = (b_1, \ldots, b_q)$. Каждой букве a_i алфавита А ставится в соответствие $B_i = \varphi(a_i)$ – слово в алфавите В длины l_i. Положим

$$Z = \sum_{i=1}^{n} q^{-l_i} = q^{-|B_1|} + q^{-|B_2|} + \ldots + q^{-|B_n|}.$$

Пусть $l = \max\limits_{i=1,\ldots,n} l_i$. Возведем Z в некоторую степень m, тогда

$$Z^m = \sum_{1 \le i_1 \dots i_m \le n} q^{-(|B_{i_1}|+|B_{i_2}|+\dots+|B_{i_m}|)} = \sum_{t=1}^{ml} S(n,t) q^{-t} .$$

Здесь введено обозначение $S(n, t)$ – число слов длины t и вида $B_{i1}B_{i2}\dots B_{in}$. Из дешифруемости следует, что все такие слова **различны**, но число различных слов длины t в алфавите из q букв не превосходит q^t , поэтому

$S(n, t) \le q^t$.

Тогда $Z^m \le \sum_{i=1}^{ml} q^t q^{-t} = ml$, где m – натуральное, а $l \ge 0$, целое. Это неравенство должно выполняться для любого m, но это возможно только тогда, когда Z не превосходит единицы, т.е.

$$Z^m \le ml \Rightarrow Z \le 1 \Rightarrow \sum_{i=1}^{n} q^{-l} \le 1.$$

Теорема доказана.

Напомним, что слово α является подсловом слова β, если существуют слова γ и δ (возможно пустые) такие, что β=γαδ. Если γ- пустое слово, то слово α называется префиксом слова β.

В общем случае из неравенства Крафта не следует дешифруемость.

Определение. Кодирование называется префиксным, если ни одно из кодовых слов не является началом другого.

Теорема. Префиксный код дешифруем.

Доказательство: Пусть есть α и β – два слова: $\varphi(\alpha) = B_{i_1} B_{i_2} \dots B_{i_n}, \varphi(\beta) = B_{j_1} B_{j_2} \dots B_{j_n}$).

Покажем, что из равенства $\varphi(\alpha) = \varphi(\beta)$ следует равенство $\alpha = \beta$.

Пусть $B_{i_1} B_{i_2} \dots B_{i_n} = B_{j_1} B_{j_2} \dots B_{j_n}$).

Поочередно слева направо сравниваем подслова слов α и β, используя знание функции ϕ, которая определяет наше побуквенное кодирование (эта функция позволяет находить $B_i = \varphi(a_i)$ среди подслов слов α и β. Сначала сравниваем B_{i_1} и B_{j_1} , затем переходим к B_{i_2} и B_{j_2} и т.д.

Если длины слов B_{i_1} и B_{j_1} одинаковы, то сами слова должны быть одинаковыми из того, что $B_{i_1} B_{i_2} \dots B_{i_n} = B_{j_1} B_{j_2} \dots B_{j_n}$). Если длины слов разные, то одно из двух слов: B_{i_1} или B_{j_1} – начало другого, что противоречит определению префиксности.

Значит, $\alpha = \beta$.

Теорема доказана.

Из теоремы сразу следует алгоритм декодирования префиксного кода.

Алгоритм декодирования. Берем первый символ слова $\varphi(\alpha)$ и ищем однобуквенное слово среди кодовых слов. Если находим такое слово, например, $B_i = \varphi(a_i)$, то декодируем слово $B_i = \varphi(a_i)$ в букву a_i и продолжаем работать, начиная со следующего символа слова $\varphi(\alpha)$. Если не находим, то добавляем следующий символ и ищем среди кодовых слов уже двухбуквенное слово. И т.д.

В случае алфавитного кодирования можно ограничиться префиксными кодами, так как, в каком-то смысле, любой дешифруемый алфавитный (побуквенный) код можно «свести к префиксному». А существование префиксного кода полностью определяется неравенством Крафта.

Теорема (о существовании префиксного кода). Префиксный код с длинами кодов $l_1 \ldots l_n$ существует тогда и только тогда, когда выполняется неравенство Крафта

$$\sum_{i=1}^{n} q^{-l_i} \le 1 \quad (5.6).$$

Доказательство. Т.к. префиксный код дешифруем, то выполняется неравенство Крафта. С другой стороны, пусть выполняется неравенство Крафта. Покажем, что в этом случае существует префиксный код. Мы просто опишем алгоритм построения такого кода, т.е. построим отображение $B_i = \varphi(a_i)$.

Пусть $l = \max\limits_{i=1,\ldots,n} l_i$. Пусть среди чисел $l_1 \ldots l_n$ имеется S_1 единиц, S_2 двоек, S_3 троек и т.д. до S_l слов длины l. При этом некоторые из S_i могут быть нулями. Упорядочим $B_i = \varphi(a_i)$ по возрастанию длины.

$B_1 = |l_1| = 1$
$B_2 = |l_2| = 1$ — S_1 слов длины 1
…
$B_{S1} = |l_{S1}| = 1$

$B_{S1+1} = |l_{S1+1}| = 2$ — S_2 слов длины 2
…

Тогда левую часть неравенства Крафта можно представить в виде:

$$\sum_{i=1}^{n} q^{-l_i} = \underbrace{\frac{1}{q} + \frac{1}{q} + \ldots + \frac{1}{q}}_{S_1} + \underbrace{\frac{1}{q^2} + \ldots + \frac{1}{q^2}}_{S_2} + \underbrace{\frac{1}{q^3} + \ldots}_{S_3}$$

Теперь пошагово строим наш префиксный код.

Первый шаг: Любые S_1 букв в B можно использовать для кодирования слов длины. Это следует из того, что при выполнении неравенства Крафта справедливо соотношение: $S_1/q \le 1$, из которого следует, что $S_1 \le q$. Таким образом мы получили первую группу однобуквенных слов.

Второй шаг: Аналогично из неравенства Крафта получаем $S_1/q + S_2/q \le 1$. Отсюда $S_1q + S_2 \le q^2$, $S_2 \le q^2 - s_1q$. Поэтому можно взять S_2 слов длины 2 в алфавите B так, что они не начинаются со слов первой группы. Таким образом мы получили вторую группу двухбуквенных слов.

Третий шаг: На этом шаге строим группу трехбуквенных слов так, чтобы они не начинались с ранее построенных слов. Такие трехбуквенные слова в нужном количестве существуют, так как вновь из неравенства Крафта следует, что $S_1/q + S_2/q + S_3 / q \le 1$. А это означает, что $S_3 \le q^3 - qS_2 - q^2S_1$.

Таким образом проходим все l шагов и в результате получаем дешифруемый префиксный код.

Теорема доказана.

Утверждение. Код Элайеса – префиксный.

Доказательство: Рассмотрим $El\ (n)$ и $El\ (m)$. Если хотя бы одно из этих слов начинается с 1, то это либо 0, либо 1. Утверждение доказано. Пусть n, m больше 1, вспомним теорему о дешифруемости кода: $El(n) = \alpha_1 \ldots \alpha_k$, $El(\mathrm{m}) = \beta_1 \ldots \beta_S$ (для определенности $k \ge s$). Пусть количество 0, с которых начинается α, совпадает с количеством 0, с которых начинается β. (В противном случае сразу следует свойство префиксности кода.) Далее рассмотрим вторые подслова слов $El\ (n)$ и $El\ (m)$. Если они не совпадают, то префиксность кода доказана. Если совпадают, то переходим к окончаниям слов. Если они совпадают, то m=n. В противном случае n не равно m. Таким образом, при m≠n выполняется соотношение $B(n) \ne B(\mathrm{m})$.

Утверждение доказано.

Докажите в качестве упражнения следующее утверждение.

Утверждение. Код Левенштейна префиксный.

5.4. Префиксные кода и q-арные корневые деревья.

Итак, у нас есть два алфавита: $A = (a_1, \ldots, a_n)$ *и* $B = (b_1, \ldots, b_q)$.

Корневое дерево – это дерево, в котором полустепень захода корня равна нулю, а всех остальных вершин – единицы, а полустепень исхода любой вершины не превосходит q, называется q-арным корневым деревом.

Ребрам, исходящим из любой вершины можно сопоставить пометки: различные буквы из алфавита B. Любой вершине однозначно можно сопоставить слово: последовательность пометок ребер от корня к этой вершине. Висячие вершины (листья) – вершины с нулевой степенью исхода.

Легко доказать следующие два утверждения.

Утверждение. Любому префиксному коду можно сопоставить q-арное корневое дерево ровно с n листьями, которым в качестве пометок сопоставлены слова $B_1, B_2, \ldots, B_n$.

Утверждение. Любому q-арному корневому дереву с n листьями, ребра и листья которого помечены по указанным выше правилам, соответствует побуквенный префиксный код, n букв которого закодированы словами – пометками листьев дерева.

5.5. Упражнения.

1. Найти El(156).

2.Найти Lev(324).

3. Найти набор натуральных чисел, если El(x)=01100001011001001000000101001001l0.

4. Найти набор натуральных чисел, если Lev(x)=11100001111000010011110100011101.

5. Множество B бинарных слов называется префиксным, если никакое слово из B не является началом другого слова из этого же множества.

Пусть l_i – слов длины i во множестве B. Доказать (не используя неравенство Крафта!) неравенство $\sum_i l_i 2^{-i} \le 1$.

6. Проверить неравенство Крафта и построить префиксный код при q=3 с длинами слов: 1,1,3,3,4,4.

6. Лекция 6. Метрические пространства и энтропийные конструкции.

Как уже говорилось выше, понятие информации не является математическим, но играет значительную роль не только внутри самой математики, но и во многих других областях знаний благодаря своему контексту и богатым коммуникационным возможностям. Эта ситуация отнюдь не является уникальной и такие понятия как «масса» или «энергия» тоже не имеют легитимного фундамента, что не мешает им играть ту роль, которая соответствует их интуитивному «физическому» смыслу.

Если считать, что информация всегда соотносится с некоторым объектом и как-то его характеризует, то понятие объекта тоже является первичным и первым его приближением является понятие множества в наивном смысле.

Если между элементами множества установлены какие-то связи, то мы получаем первые приближения к «реальным» объектам, которые характеризуются терминами граф, гиперграф и так далее. Измерение информации — это, в каком-то смысле, измерение расстояния между самим объектом и тем образом, который мы используем для его представления.

6.1 Информация по Хартли

Мощность множества является лишь одной из его характеристик. Однако в определённых ситуациях эта характеристика играет определяющую роль и все множества одинаковой мощности лежат в одном классе эквивалентности.

Определение. Пусть M — конечное множество и $|M| = m$. Тогда число $\log_2 m$ называется *энтропией* множества M.

Это определение тесно связано со способом задания конечного множества. Для того, чтобы задать M, нужно найти способ или кодировку его элементов. Нет разницы между множеством M и свойствами элементов M. Если рассмотреть некоторое число предикатов $P_1, P_2, \dots, P_k$, заданных на M, то для того, чтобы эти предикаты различали элементы M необходимо выполнение условия: $2^k \ge m$. Действительно, каждому элементу $a \in M$

поставим в соответствие двоичный набор $(P_1(a), \dots, P_k(a))$, где $P_r(a)$ — значение предиката P_r на элементе a. Число таких наборов будет 2^k и потому для различения элементов M потребуется выполнение неравенства

$$k \geq \log_2 m.$$

Это условие имеет «информационный» характер и потому число $\log_2|M|$ носит название информации или энтропии по Хартли конечного множества M.

Примеры

1) Рассмотрим все целые числа из интервала $[0, n]$. Каждое из них может быть представлено в двоичной системе счисления. Словом длины m из нулей и единиц, где $m = [\log_2 n] + 1$. Таким образом, энтропия по Хартли этого множества равна $\log_2(n+1)$.
2) Пусть T_n — множество всех натуральных решений уравнения

$$x + y + z = n.$$

Раньше было показано, как можно найти $|T_n|$ и доказывалось, что

$$|T_n| = \binom{n+2}{2}.$$

Таким образом, энтропия по Хартли множества T_n равна $\log_2|T_n| = \log_2\binom{n+2}{2} \sim 2\log_2 n$. Однако «описание» множества T_n является отдельной задачей и $H(T_n)$ может служить лишь оценкой сложности такого описания.

6.2 Энтропийные «рассуждения»

Существует тип доказательных рассуждений в основе которых лежат *энтропийные* или, что то же самое, *мощностные* соображения. Вместо их описаний мы приведём несколько примеров.

Пример. Даны n натуральных чисел $a_1, a_2, \dots, a_n$. Показать, что из них можно составить сумму, делящуюся на n.

Доказательство. Рассмотрим следующие суммы

$$S_1 = a_1, S_2 = a_1 + a_2, \dots, S_n = a_1 + a_2 + \dots + a_n.$$

Так как число вычетов по $mod\ n$ равно n, то либо какая-то из S_k делится на n, либо $S_{k+i} \equiv S_k\ (mod\ n)$. Последнее означает, что сумма $S_{k+i} - S_k \equiv 0\ (mod\ n)$. Что и требовалось доказать.
Пример. В квадрат T со стороной n брошена (n^2+1) точка. Доказать, что среди этих точек найдётся пара с расстоянием $\leq 2^{1/2}$.

Доказательство. Разобьём квадрат T на n^2 квадратов со стороной 1. Тогда хотя бы один из этих квадратов содержит ≥ 2 точек из брошенных. Расстояние между этими точками $\leq 2^{1/2}$.
Пример. Рассмотрим конъюнктивную нормальную форму (К.Н.Ф.) $or = or_1 \cdot or_2 \cdot \dots \cdot or_m$, где

$$or_k = \left(x_{1k}^{G_1} \vee x_{2k}^{G_2} \vee \dots \vee x_{rk}^{G_k}\right), \qquad k = \overline{1, m}.$$

Проблема состоит в поиске $x \in B^n$ такой, что $or(x) = 1$. Число «невыполняющих» наборов для or_k равно 2^{n-r_k}, так как число решений уравнения

$$x_{1k}^{G_1} \vee x_{2k}^{G_2} \vee \ldots \vee x_{rk}^{G_k} = 0$$

равно 2^{n-r_k}. Поэтому общее число «невыполняющих» наборов для К.Н.Ф. or не превосходит $m2^{n-r_k}$. Отсюда вытекает следующее утверждение.

Утверждение. Если каждая скобка в or содержит больше, чем $\log_2 m$ букв, то К.Н.Ф. $or = or_1 \cdot or_2 \cdot \ldots \cdot or_m$ выполнима.

Доказательство. Если $N = \min r_k$, то число невыполняющих наборов для or не превосходит $m2^{n-N}$. Так как $m2^{n-N} < 2^n$ при $N > \log_2 m$, то or выполнима.

Замечание. Неравенство $\sum_{k=1}^{m} 2^{n-r_k} < 2^n$ является достаточным для выполнимости К.Н.Ф. Об «информационном» смысле этого неравенства речь будет идти ниже.

Пример. Следующее «энтропийное» рассуждение устанавливает бесконечность множества простых чисел путём «подсчёта возможностей».

Так как каждое натуральное число $N \leq n$ имеет единственное представление в форме $N = p_1^{\alpha_1} \ldots p_k^{\alpha_k}$, где $k \leq \pi(n)$ и $\alpha_i \leq \log_2 N \leq \log_2 n$, то N может быть «закодировано» набором длины $\pi(n)$ натуральных чисел $(\alpha_1, \alpha_2, \ldots, \alpha_{\pi(n)})$, где $\alpha_i \leq \log_2 n$. Число таких наборов не превосходит $(\log_2 n)^{\pi(n)}$ и потому должно быть справедливо неравенство $(\log_2 n)^{\pi(n)} > n$ и потому $\pi(n) \geq \frac{\log_2 n}{\log_2 \log_2 n}$. Отсюда следует, что $\lim_{n\to\infty} \pi(n) = \infty$.

Что и требовалось доказать.

6.3 ϵ-сети.

Хотя понятие энтропии по Хартли было введено только для конечных множеств, существуют математические конструкции, которые позволяют распространить это определение для бесконечных множеств, обладающих определённой структурой, дающей возможность «аппроксимировать» элементы этих множеств с помощью понятия расстояния.

Ключевым понятием при этом является понятие ϵ-сети, и энтропия по Хартли — это энтропия этой ϵ-сети, которая является конечным множеством при определённых ограничениях на исходное бесконечное множество.
Пусть V — некоторое метрическое пространство с метрикой ρ и $S \subseteq V$. Тогда пара (S, ρ) называется метрическим множеством.

Определение. Множество S_ϵ называется ϵ-сетью для S, если любой элемент из S лежит в шаре радиуса $\leq \epsilon$ с центром в некоторой точке $x \in S_\epsilon$.

Если минимальная мощность ϵ-сети равна m_ϵ, то величина $\log_2 m_\epsilon = H_\epsilon(S)$ называется ϵ-*энтропией* множества S. В определённом смысле ϵ-энтропия характеризует «*сходство*» элементов множества $S \subseteq V$.

Следующее определение связано с понятием *различимость*.

Определение. Множество $R \subseteq S$ называется , если ϵ- *различимым*

$$\rho(x, y) > \epsilon$$

для $x, y \in R$.

Рассмотрим все ϵ-различимые множества в S и через $h_\epsilon(S)$ обозначим энтропию максимального по мощности ϵ-различимого множества в S. Следующие простые неравенства связывают ϵ-энтропию и или $h_\epsilon(S)$.

Утверждение. Для всякого ограниченного и замкнутого множества $S \subseteq V$ справедливы неравенства

$$h_{2\epsilon}(S) \le H_\epsilon(S) \le h_\epsilon(S). \quad (6.1)$$

Доказательство. Если R — максимальное по мощности ϵ-различимое множество, то шары радиуса ϵ с центром в R покрывают S. Действительно, если точка x_0 не покрыта, то $\rho(x_0, y) > \epsilon$ для всех y из R. Но тогда множество $R' = R \cup \{x_0\}$ — это ϵ-различимое множество, что противоречит определению R. Отсюда следует верхняя граница

$$H_\epsilon(S) \le h_\epsilon(S).$$

С другой стороны, в каждом шаре радиуса ϵ не может быть более одной точки из 2ϵ-различимого множества. Справедливо неравенство

$$h_{2\epsilon}(S) \le H_\epsilon(S).$$

Что и завершает доказательство сформулированного утверждения.

Использование неравенств (6.1) является стандартным приёмом при оценке ϵ-энтропии метрических пространств. Ниже приведено несколько примеров таких вычислений.

Пример. Пусть E^n — n-мерное пространство над полем вещественных чисел с метрикой Эвклида и $S = B^n$ — единичный n-мерный куб

$$B^n = \{x = (x_1, \dots, x_n):\ 0 \le x_i \le 1,\ i = \overline{1, n}\}.$$

Найдём ϵ-энтропию множества B^n.

Рассмотрим следующую решётку в B^n

$$C_\epsilon = \{(i_1\epsilon, i_2\epsilon, \dots, i_n\epsilon)\},$$

где $0 \le i_k \le \left[\frac{1}{\epsilon}\right], k = \overline{1, n}$.

Мощность решётки C_ϵ равна $\left(\left[\frac{1}{\epsilon}\right] + 1\right)^n$ и для $x, y \in C_\epsilon$ справедлива оценка

$$\rho(x, y) = \sqrt{\sum_{i=1}^{n} \epsilon^2 (i_k - i'_k)^2} = \epsilon \sqrt{\sum_{i=1}^{n} (i_k - i'_k)^2} \ge \epsilon.$$

Таким образом, C_ϵ — это ϵ-различимое множество в B^n, имеющее энтропию

$$n\log_2\left(\left[\frac{1}{\epsilon}\right] + 1\right) \sim n\log_2\frac{1}{\epsilon} \quad (6.2)$$

при $\epsilon \to 0$. Отсюда получаем оценку

$$H_\epsilon(B^n) \geq h_{2\epsilon}(B^n) \geq \log_2|C_{2\epsilon}| \sim n\log_2\frac{1}{2\epsilon} \gtrsim n\log_2\frac{1}{\epsilon}.$$

Пусть теперь $x = (x_1, \dots, x_n) \in B^n$. «Приблизим» точку x следующей точкой решётки C_ϵ: $x' = (x'_1, x'_2, \dots, x'_n)$, где

$$x'_k = \left[\frac{x_k}{\epsilon}\right]\epsilon,\ k = \overline{1,n}.$$

Имеем, по определению,

$$x_k - \epsilon \leq x'_k \leq x_k$$

и отсюда следует оценка

$$\rho(x, x') \leq \sqrt{n\epsilon^2} = \epsilon\sqrt{n}. \quad (6.3)$$

Из неравенства (6.3) следует, что точки решётки C_ϵ образуют $(\epsilon\sqrt{n})$-сеть в B^n. Отсюда и из (6.1) получаем оценку

$$H_{\epsilon\sqrt{n}}(B^n) \lesssim n\log_2\frac{1}{\epsilon}$$

и потому

$$H_\epsilon(B^n) \lesssim n\log_2\frac{\sqrt{n}}{\epsilon\sqrt{n}} \sim n\log_2\frac{1}{\epsilon}.$$

Отсюда вытекает

$$H_\epsilon(B^n) \sim n\log_2\frac{1}{\epsilon}.$$

Пример. В качестве второго примера обратимся к вычислению ϵ-энтропии множества дифференцируемых функций с ограниченной производной, заданных на отрезке $[0, l]$. Это множество мы обозначим через $F(l, d)$, где

$$|f'(x)| < d$$

при $f(x) \in F(l, d)$.

Разобьём отрезок $[0, l]$ на интервалы длины δ. Число точек в полученной решётке равно $\left[\frac{l}{\delta}\right]$. Рассмотрим разложение функции $f(x)$ в ряд Тейлора в точке решётки a:

$$f(x) = (x - a) + \frac{f'(x)}{1!}(x - a) + R_1(x),$$

где $R_1(x)$ — остаточный член в какой-нибудь классической форме. Мы используем известную оценку

$$|f(b) - f(a)| \leq \max_c f'(c)(b - a),$$

где $c \in [a, b]$. Для нашего случая $x \in [0, l]$ и $|x - a| < \delta$. Поэтому

$$|f(x) - f(a)| < d\delta.$$

Если положить $d\delta = \epsilon$, то получим неравенство

$$|f(x) - f(a)| < \epsilon.$$

Число точек решётки при таком выборе параметров равно $\frac{l}{\delta} = \frac{dl}{\epsilon}$ и потому

$$H_\epsilon \sim \log_2 \frac{1}{\epsilon}.$$

Более общим утверждением типа сформулированного выше является следующая теорема А.Н. Колмогорова.

Пусть $F(n, \rho, B)$ — множество функций от n-переменных, имеющих частные производные до порядка ρ включительно, где $\rho \geq 1$, ограниченные сверху константой B. Если это множество снабдить чебышевской метрикой

$$\rho(f_1, f_2) = \sup_{x \in K} |f_1(x) - f_2(x)|$$

при $x = (x_1, x_2, \dots, x_n) \in K$ и обозначить через $H_\epsilon(n, \rho, B)$ его ϵ-энтропию, то имеет место следующее утверждение.

Теорема А.Н. Колмогорова. Справедлива асимптотическое соотношение

$$H_\epsilon(n, \rho, B) \approx C(n, \rho, B) \left(\frac{1}{\epsilon}\right)^{\frac{n}{\rho}},$$

где $C(n, \rho, B)$ — константа, не зависящая от ϵ.

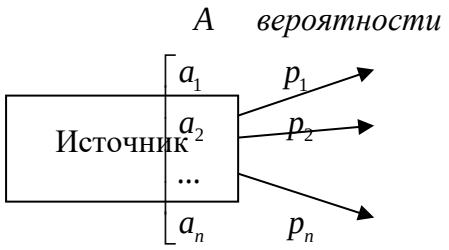
Стандартная интерпретация соотношения (7.6) следующая: сложность класса функций $F(n, \rho, B)$ разумно измерять величиной отношения $\frac{n}{\rho}$. Последнее, в частности, означает следующее: если $\frac{n}{\rho} > \frac{n_1}{\rho_1}$, то существуют функции из класса $F(n, \rho, B)$, не представимые в виде суперпозиции функций из класса $F(n_1, \rho_1, B_1)$.

6.4 Сжатие информации. Энтропия по Шеннону.

Этот теоретический предел возможного сжатия и определяется *количеством информации* в том *слове* (*множестве слов*) с помощью которых *информация* представлена.

6.4.1 Математическая модель: алфавитное кодирование случайного источника.

Рассмотрим следующую модель. Имеется источник, который поочередно генерирует буквы алфавита A=($a_1 \dots a_n$) с вероятностями $p_1 \dots p_n$. То есть имеет место следующая ситуация.



$$\sum p_i = 1$$
$$p_i \geq 0$$

Каждая буква a_i генерируется с некоторой вероятностью p_i. Эта вероятность p_i не зависит от того, что выдал источник ранее и что он будет выдавать потом. Она также не зависит от очередности выдачи букв источником и от того, какой по счету от начала выдачи выдана данная буква. Такой источник называется *источником Бернулли*, что является частным случаем *стационарного источника*. На практике встречаются и другие источники. Для них приведенный ниже результаты неверны, но могут быть получены их аналоги.

6.4.2 Энтропия по Шеннону

Опр. Энтропией случайного источника (энтропией Шеннона) называется число

$$H(A) = \sum_{i=1}^{n} p_i \log \frac{1}{p_i}.$$

6.4.3 Энтропия по Шеннону и энтропия по Хартли.

Заметим, что Энтропия по Хартли является частным случаем энтропии по Шеннону. Если все $p_i = \frac{1}{n}$, то $H(A) = \log n$, а это и есть Энтропия по Хартли множества из n элементов. В общем случае справедливо следующее утверждение, которое показывает, что энтропия по Хартли является оценкой сверху для энтропии по Шеннону. То есть ситуация равновероятных событий наихудшая с точки зрения информативности.

В следующем и других доказательствах нам потребуется математический факт, известный как неравенство Йенсена.

Неравенство Йенсена. Пусть даны числа $\alpha_1 \ldots \alpha_n$, $\alpha_i > 0$, $\sum \alpha_i = 1$. Тогда для выпуклой вверх функции f(x) справедливо неравенство.

$$\sum_{i=1}^{n} \alpha_i f(x_i) \le f(\sum_{i=1}^{n} \alpha_i x_i).$$

Для выпуклой вниз функции выполняется противоположное неравенство:

$$\sum_{i=1}^{n} \alpha_i f(x_i) \ge f(\sum_{i=1}^{n} \alpha_i x_i).$$

Теорема. Пусть имеется случайный источник, который генерирует буквы алфавита $A = (a_1 \ldots a_n)$ с вероятностями $p_1 \ldots p_n$,тогда справедливо соотношение

$$H(A) = \sum_{i=1}^{n} p_i \log \frac{1}{p_i} \le \log n.$$

Доказательство. Рассмотрим функцию $f(x) = -x \log x$. Так как $f'(x) = -\log x - 1/\ln 2$ и $f''(x) = -1/x$, то при $x \ge 0$ функция $f(x)$ выпукла вверх. Тогда для нее справедливо неравенство Йенсена: $\sum_{i=1}^{n} \alpha_i f(x_i) \le f(\sum_{i=1}^{n} \alpha_i x_i)$. Возьмем $\alpha_1 \ldots \alpha_n$, $\alpha_i > 0$, $\sum \alpha_i = 1$

Положим $\alpha_i = 1/n$, $x_i = p_i$,i=1,…,n.

$$H(A) = \sum_{i=1}^{n} p_i \log \frac{1}{p_i} = n \sum_{i=1}^{n} \frac{1}{n} f(p_i) \le n f\left(\sum_{i=1}^{n} \frac{p_i}{n}\right) = -n\left(\sum \frac{p_i}{n}\right) \log\left(\sum_{i=1}^{n} \frac{p_i}{n}\right) = \log n$$

Здесь мы учли, что $\sum_{i=1}^{n} \frac{p_i}{n} = 1$/n.

Теорема доказана.

6.4.4 Теорема Шеннона

Рассмотрим теперь математическую модель, являющуюся объединением двух предыдущих. Случайный источник генерирует буквы алфавита $A = (a_1, \ldots, a_n)$, к которым потом применяется некоторый алгоритм алфавитного кодирования φ. (Каждой букве a_i

алфавита A ставится в соответствие $B_i = \varphi(a_i)$ – слово длины l_i в алфавите $B = (b_1, \dots, b_q)$). Как можно оценить качество такого алгоритма?

В случае вероятностного источника такие количественные характеристики алгоритма как $\sum_{i=1}^{n} l_i$ или $\frac{1}{n}\sum_{i=1}^{n} l_i$ уже не являются информативными, так как не зависят от вероятностей.

В этом случае удобно использовать величину $C_\phi(A) = \sum_{i=1}^{n} p_i l_i$, которая характеризует среднее количество букв алфавита B , приходящееся на один символ алфавита A при алфавитном кодировании φ. (Назовем эту величину стоимостью кодирования φ). Пусть теперь $C(A) = \min_{\phi} C_\phi(A)$, где минимум берется по всем возможным дешифруемым алгоритмам алфавитного кодирования (при заданных алфавитах A и B и распределении вероятностей $p_1 \dots p_n$). (Назовем эту величину минимальной стоимостью кодирования).

Если бы мы знали C(A), то смогли бы оценить качество нашего алгоритма φ.

Следующая теорема (она известна под названием теоремы Шеннона для канала без шума) дает ответ на этот практически важный вопрос.

Теорема. Справедливо соотношение

$$\frac{H(A)}{\log q} \le C(A) \le \frac{H(A)}{\log q} + 1.$$

Доказательство.

Сначала докажем оценку снизу:

$$\frac{H(A)}{\log q} \le C(A).$$

В следующих выкладках все суммы берутся по i=1,..,n. Используется известное из математического анализа неравенство $\ln x \le x - 1$, $\log x \le (x-1)\log e$, а также неравенство Крафта (суммирование везде ведется от 1 до n).

$$\sum p_i \log \frac{1}{p_i} - \log q \sum p_i l_i = \sum p_i \left(\log \frac{1}{p_i} - l_i \log q \right) = \sum p_i \log \frac{q^{-l_i}}{p_i} \le \log e \sum p_i \left(\frac{q^{-l_i}}{p_i} - 1 \right) (=)$$

$$(=) \log e \left[\sum q^{-l_i} - \sum p_i \right] \le 0 \qquad \left(\sum q^{-l_i} \le 1 \text{ – по неравенству Крафта, } \sum p_i = 1 \right).$$

Для доказательства оценки сверху мы построим конкретный алгоритм алфавитного кодирования ϕ (известный как алгоритм Шеннона-Фано). Ведь если оценка сверху будет выполняться для некоторого $C_\phi(A)$, то она будет выполняться и для C(A).

В данном коде набор длин задается следующим соотношением.

$l_i = \left\lceil \frac{\log \frac{1}{p_i}}{\log q} \right\rceil$ (целая часть с избытком (сверху)).

Проверим, удовлетворяет ли этот набор неравенству Крафта.

$\sum q^{-l_i} \le 1$, $\sum q^{-\left\lceil \frac{\log \frac{1}{p_i}}{\log q} \right\rceil} \le \sum q^{-\frac{\log \frac{1}{p_i}}{\log q}} = \sum q^{-\log_q \frac{1}{p_i}} = \sum p_i = 1 \Rightarrow$ неравенство выполняется.

Теперь построим код точно также, как это делалось выше при доказательстве теоремы о существовании префиксного кода в случае выполнения неравенства Крафта (мы строили сначала слова B_i длины 1, затем слова длины два и т.д.).

Оценим теперь сверху величину $C_\phi(A)$ для этого кода.

$C_\phi(A) = \sum p_i l_i \le \sum p_i \left\lceil \frac{\log \frac{1}{p_i}}{\log q} \right\rceil \le \sum p_i \left(\frac{\log \frac{1}{p_i}}{\log q} + 1 \right) = \frac{H(A)}{\log q} + 1$.

Теорема доказана.

В наших примерах часто будет использоваться двоичный алфавит. Очевидно, что для случая бинарного алфавита B получаем неравенство

$$H(A) \le C(A) \le H(A) + 1.$$

7. Лекция 7. Свойства энтропии по Шеннону. Алгоритмы сжатия информации.

7.1 Свойства энтропии.

Здесь в качестве справки мы приведем некоторые свойства энтропии по Шеннону. Это даст вам возможность лучше понять смысл этого понятия и сферу его практического применения. Для дальнейшего изложения нам понадобятся только свойства 5 и 6. Они очевидны из определения энтропии и простейших комбинаторных соотношений, известных вам и могут быть никак не связаны с *Теорией вероятностей*. В качестве их следствия получается свойство 7. Остальные утверждения приводятся для тех, кто знаком с самыми азами этой теории.

Случайный источник можно трактовать как событие с множеством исходов, вероятности которых заданы.

Рассмотрим два события X и Y. Исходы первого события будем обозначать через x, а второго - через y. Введем по определению понятие условной энтропии H(X/y).

$H(X|y) = -\sum_x p(x|y) \log p(x|y)$.

Это случайная величина. А теперь усредним $H(X|y)$ по всему множеству Y и получим уже неслучайную величину $H(X|Y)$.

$$H(X/Y) = -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(x|y).$$

Перейдем теперь к перечислению свойств. В некоторых случаях приводятся доказательства.

1) $H(\mathrm{X}) \geq 0$.

2) $H(\mathrm{X})$ является выпуклой вверх функцией своих аргументов функция своих аргументов $p_1 \ldots p_n$.

3) $H(\mathrm{X}) \leq \log|\mathrm{X}|$ (если все события равновероятны, то энтропия максимальна).

4) $H(X|Y) \leq H(\mathrm{X})$ (Аналог свойства 3 для условной энтропии. Дополнительная информация не увеличивает энтропию).

$$H(X/Y) - H(X) = -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(x|y) + \sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(x) =$$

$$= \sum_{x\in X}\sum_{y\in Y} p(x,y)\log(p(x)/p(x|y)) \leq \sum_{x\in X}\sum_{y\in Y} p(x,y)(p(x)/p(x|y)-1)\log e =$$

$$= \log e(\sum_{x\in X}\sum_{y\in Y} p(y)p(x) - \sum_{x\in X}\sum_{y\in Y} p(x,y)) = 0.$$

5) Пусть X и Y – независимые события. Справедливо равенство

$H(XY) = H(X) + H(Y)$.

6) Пусть два события имеют равное число исходов, а вероятности этих исходов как множества совпадают, т.е. они состоят из одних и тех же $(p_1,\ldots,p_n)$, но в разном порядке. Справедливо равенство

$H(\mathrm{X}) = H(\mathrm{X}')$.

7) Пусть есть событие $X = (p_1,\ldots,p_n)$ и некоторое подмножество его исходов $A \subseteq (p_1,\ldots,p_n)$. Построим новое событие X_A таким образом, что его исходы задаются вероятностями

$$p_i^A = \begin{cases} \dfrac{\sum_A p_i}{|A|}, \text{ если } p_i \in A \\ p_i, \text{ если } p_i \notin A \end{cases}$$

(«сглаживание» фрагмента).

Справедливо неравенство

$H(x) \leq H(X_A)$.

Доказательство. Доказательство следует из свойств следует из (2) и (6).

По свойству (2) энтропия выпуклая вверх функция своих аргументов, а для такой функции f выполняется неравенство Йенсена $f\left(\sum \alpha_i x_i\right) \geq \sum \alpha_i f(x_i)$.(Здесь числа $\alpha_1 \ldots \alpha_n$, $\alpha_i > 0$, $\sum \alpha_i = 1$.)

Без ограничения общности считаем, что первые m элементов X образуют подмножество A, т.е. исходы X задаются вероятностями $p_1 \ldots p_m p_{m+1} \ldots p_n$

Строим вспомогательные события:

$$\begin{cases} X^1 = X \\ X^2 = p_2 p_3 \ldots p_m p_1\ p_{m+1} \ldots p_n \\ X^3 = p_3 p_4 \ldots p_1 p_2\ p_{m+1} \ldots p_n \\ \quad\vdots \\ X^m = p_m p_1 \ldots p_{m-2} p_{m-1}\ p_{m+1} \ldots p_n \end{cases}$$

Из свойства (6) следует, что $H(\mathrm{X}) = H(\mathrm{X}^1) = \ldots = H(\mathrm{X}^m)$ (все события состоит из одинаковых множеств значений вероятностей). Тогда можно X_A представить в виде $(H(\mathrm{X}) + H(\mathrm{X}^1) + \ldots + H(\mathrm{X}^m))/\mathrm{m}$. Применяем неравенство Йенсена, где все $\alpha_i = 1/m$. Получаем $\mathrm{H}(X_A) \geq (H(\mathrm{X}) + H(\mathrm{X}^1) + \ldots + H(\mathrm{X}^m))/\mathrm{m}$. Отсюда следует$\mathrm{H}(X_A) \geq \mathrm{H}(\mathrm{A})$.

8) $H(XY) = H(X) + H(Y|X) = H(Y) + H(X|Y)$.

Следует из известных в теории вероятностей соотношений: $P(XY) = P(X) \cdot P(Y|X)$ и $P(XY) = P(Y) \cdot P(X|Y)$.

9) $H(X_1 X_2 \ldots X_n) = H(X_1) + H(X_2|\ X_1) + H(X_3|\ X_1\ X_2) + \ldots + H(X_n|\ X_1\ X_2 \ldots\ X_{n-1})$.

Следует из известных в теории вероятностей соотношения: $P(X_1 \ldots X_n) = P(X_1) \cdot P(X_2|X_1) \cdot \ldots \cdot \mathrm{P}(\mathrm{X_n}|\mathrm{X_1},\ldots,\mathrm{X_{n-1}})$.

10) $H(X|YZ) \leq H(X|Y)$.

Доказывается по аналогии с доказательством свойства (4).

11) Пусть задано событие X и на множестве его исходов определена функция $y = g(x)$. Введем событие Y с множеством исходов $y = g(x)$. Тогда $H(Y) \leq H(X)$. Равенство $H(Y) = H(X)$ достигается тогда и только тогда, когда существует обратная функция g^{-1}.

Доказательство. Следует из свойства (8).

$H(X) + H(Y|X) = H(Y) + H(X|Y) \Rightarrow H(Y) = H(X) + H(Y|X) - H(X|Y) \geq 0$.

Т.к. y полностью определяется по x, то $H(Y|X) = 0$.

7.2 Алгоритмы кодирования

7.2.1 Алгоритм Шеннона (Фано).

Передаваемые символы располагаются в порядке убывания (или возрастания) вероятностей. Построим таблицу, первым столбцом которой являются идентификаторы передаваемых символов. Назовем, например, верхней частью этой получившейся

последовательности символов ту, которая начинается минимального номера. Эта упорядоченная последовательность фиксируется, затем идут шаги алгоритма.

Каждый шаг – это горизонтальное деление на q частей этой последовательности, а i-й шаг – деление одной из выбранных на предыдущем этапе частей. Рассмотрим частный случай бинарного алфавита. В этой ситуации деление происходит на две части.

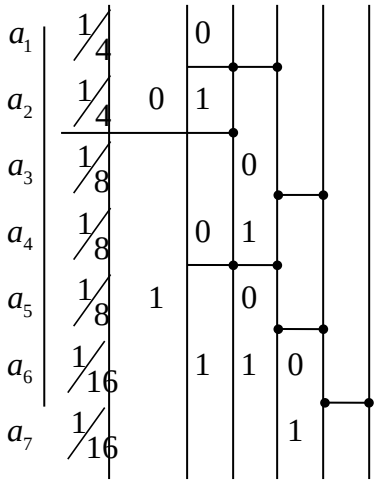
Деление происходит так, чтобы в каждой половине суммы вероятностей отличались бы на минимально возможную величину (при нескольких вариантах берется, например, с избытком часть с наименьшими номерами букв).

Затем каждое деление порождает символ алфавита B, который приписывается при таком делении к коду верхней части добавляется 0, а нижней – 1.

Так происходит до тех пор, пока в каждой из получившихся частей не останется по одному элементу. Тогда приписанные части последовательности нулей и единиц и будут кодами этих элементов.

Пример.

Нужно передавать семь букв. Их вероятности заданы. Схема алгоритма приведена ниже.



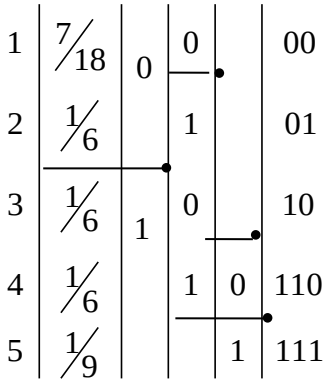
Таким образом полученные коды букв такие:

a_1 – 00, a_2 – 01, a_3 -100, a_4 -101, a_5 -110, a_6 -1110, a_7 -1111.

Стоимость кодирования 2,625. Энтропия:

H = 1/4 log 4 + 1/4 log 4 + 1/8 log 8 + 1/8 log 8 + 1/8 log 8 + 2 * 1/16 log 16 = 2, 25

Пример.



Стоимость кодирования 2,28.

H = 7/18 log 18/7 + 1/2 log 6 + 1/5 log 9 = 2,17

7.2.2 Алгоритм Хаффмана

Алгоритм дает результат не хуже, чем метод Шеннон-Фано. Более того, можно показать, что стоимость кодирования этим алгоритмом равна C(A).

Для случая двоичного кодирования код строится при помощи бинарного корневного дерева. (В случае произвольного q используется q-арное дерево). Из каждой вершины выходит два ребра: правое помечаем 1, левое – 0. Листья такого дерева кодируются последовательностью пометок от корня к листу. Очевидно, что множество соответствующих этим листьям слов образует префиксный код.

Вначале мы строим вершины дерева (листья), соответствующие передаваемым буквам. Они помечаются значениями вероятностей. Затем на каждом шаге берутся две вершины с наименьшими пометками. Они образуют родительский узел, пометка которого равна сумме пометок вершин, и исключаются из дальнейшего рассмотрения. Шаги продолжаются до появления корня.

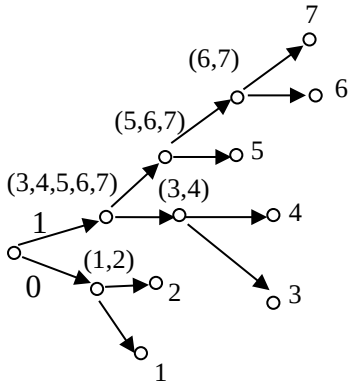
Пример.

Рассмотрим тот же пример, что и для кода Шеннона-Фано.

Итерации о построение листьев бинарного дерева.

(1, 2, 3, 4, 5, 6, 7) → (1, 2, 3, 4, 5, (6, 7)) → (1, 2, (3, 4), 5, (6, 7)) → (1, 2, (3, 4), (5, 6, 7)) → ((1, 2), (3, 4), (5, 6, 7)) → ((1, 2), (3, 4, 5, 6, 7))

Теперь, начиная с корня, строим пометки ребер дерева:



Получаем код. a_1 – 00, a_2 – 01, a_3 -100, a_4 -101, a_5 -110, a_6 -1110, a_7 -1111.

Стоимость кодирования 2,625.

7.2.3 Блочное кодирование Хаффмена.

Алгоритм Хаффмена применяется не к самим буквам а к их блокам Тогда теорема Шеннона распространяется уже не буквы, а на блоки. В качестве вероятности блока берется произведение вероятностей, входящих в него букв. Если стоимость блока поделить на количество букв в нем, то «получим стоимость кодирования на одну букву».

Пример.

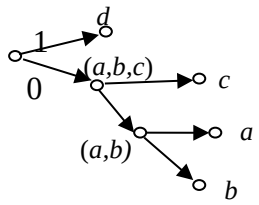
Пусть случайный источник генерирует 0 с вероятностью 0,25 и единицу с вероятностью 0,75. Стоимость кодирования равна 1 при энтропии 0,81.

Если поток из 0 и 1 будем интерпретировать как пары (блоки из двух букв), то это будет равносильно необходимости передавать четыре буквы с вероятностями:

a	00	1/16	000
b	01	3/16	001
c	10	3/16	01
d	11	9/16	1

Применяем алгоритм Хаффмена:

$(a, b, c, d) \rightarrow ((a, b), c, d) \rightarrow (a, b, c), d)$



Стоимость на блок $l\,(xx) = \sum p_i\, l_i = 3/16 + 9/16 + 6/16 + 9/16 = 27/16$.

Стоимость на одну букву $l(x) = 27/32 = 0,844$.

Если перейти теперь к блокам длины три, то получим:

a 000 1/64 00000

b 001 3/64 00001

c 010 3/64 00010

d 011 9/64 010

e 100 3/64 00011

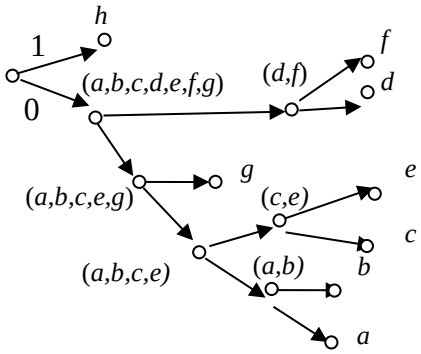
f 101 9/64 001

g 110 9/64 001

h 111 27/64 1

Алгоритм Хаффмена:

(a, b, c, *D*, *e*, f, g, *H*) → ((a, b), c, *D*, *e*, f, g, *H*) → ((a, b), (c, *e*), *D*, f, g, *H*) → ((a, b, c, *e*), *D*, f, g, *H*) → ((a, b, c, *e*), (*D*, f), g, *H*) → ((a, b, c, *e*, g), (*D*, f), *H*) → ((a, b, c, *D*, *e*, f, g), *H*)



Стоимость кодирования на блок *l(xxx)* = 5/64 + 15/64 + 15/64 + 15/64 +28/64 + 15/64 +27/64 +27/64 +27/64 = 50/64 +108/64 = 158/64 = 2,46875.

В переводе на одну букву получаем l(x)=0,823.

Процесс можно продолжать.

Ниже будет показано, что при неограниченном росте длины блока стоимость кодирования на одну букву может быть сколь угодно близка к значению энтропии.

7.2.4 Алгоритм арифметического кодирования

Появление этого алгоритма было связано с попыткой обойти сложности кодирования (декодирования) блочного алгоритма. Однако сравнивать его с предыдущими алгоритмами математически некорректно. Дело в том, что арифметическое кодирование не дает префиксного кода и, вообще, оно не переводит двоичные слова в двоичные слова. В нем используется третий символ – разделитель (обозначим его *). Уже на канальном уровне модели OSI приемник различает его от 0 и 1.

Поэтому сравнивать стоимость арифметического кода, например, в блочным кодом Хэмминга – некорректно, т.к. все зависит от стоимости разделителя. Очевидно, что, если его передача стоит много дороже передачи бита, то использование его нецелесообразно. Обратно, если его передача стоит много дешевле передачи бита, то математическая задача перешла в инженерную.

Единственная почва для сравнения – считать, что разделитель , с точки зрения стоимости, эквивалентен биту.

Принципы арифметического кодирования заключаются в следующем:

1) K - кодовое слово. Ему ставится в соответствие точка $(\bullet) \in [\,0,\ 1\,]$

2) Строится разбиение отрезка $[\,0,\ 1\,]$ на совокупность полуинтервалов $W_1,\ldots,W_s$:

$$[\,0,\ 1\,] = \bigcup_{i=1}^{s} W_i : \begin{cases} а)\ W_i \cap W_j = \varnothing \\ б)\ \bigcup_{i=1}^{s} W_i = [0,1] \end{cases}$$

3) Длина W_i равна «вероятности» слова, закодированного точкой этого отрезка.

$$\frac{k}{2^p} \Rightarrow \begin{cases} 1)\ \dfrac{k}{2^p} \in W_i \\ 2)\ k - \textit{натуральное} \\ 3)\ p - \textit{наименьшее} \cdot \textit{возможное} \end{cases}$$

4) Точка $T_i \in W_i$ имеет вид

Кодовое слово имеет p разрядов и кодируется в двоичную запись числителя k. (Эта двоичная запись не может иметь больше, чем p разрядов, а если разрядов меньше, чем p, то добавляется префикс из нулей). Таким образом, при одинаковых числителях слова будут различаться количеством разрядов.

Так как разбиение отрезка и точки на нем известны кодеру и декодеру, то алгоритм декодирования очевиден.

7.3 Блочное кодирование и теорема Шеннона.

Используя свойства энтропии можно, в каком-то смысле «уточнить» теорему Шеннона. Пусть мы будем кодировать не буквы алфавита A, а блоки из k букв алфавита A. Вероятность каждого блока – это произведение вероятностей входящих в него букв. Тогда блок может быть представлен как произведение независимых событий. Энтропию такого блока H(AA…A) обозначим H^k(A). Минимальную стоимость кодирования для блока обозначим через C^k(A). Тогда стоимость кодирования C(A) одной буквы блока можно принять за C(A)= C^k(A)/k.

Из свойства энтропии (5) следует, что H(A)= H^k(A)/k. Из этих соотношений и теоремы Шеннона для блоков получаем

C^k(A) – 1 ≤ H^k(A)/logq ≤C^k(A)

Отсюда следует, что

C^k(A)/k – 1/k ≤ H(A)/logq ≤C^k(A)/k.

Отсюда при k →∞ получаем, что H(A)/logq → C(A).

За такую точность приходится платить трудоемкостью алгоритмов кодирования (декодирования).

Таким образом, была доказана теорема.

Теорема. Для любого $\varepsilon > 0$ существует N_ε такое, что при блочном кодировании с числом блоков $N > N_\varepsilon$ справедливо неравенство $|C_\varepsilon(A) - H(A)| < \varepsilon$.

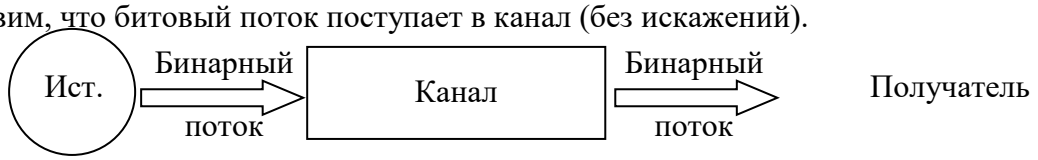
Здесь C_ε(A) - среднее число символов на одну букву в данном алгоритме кодирования при данном $\varepsilon > 0$. (Например, блочный алгоритм Хаффмена.)

Данное утверждение известно как теорема Шеннона для канала без шума. Приведем это утверждение в более распространенной формулировке.

Пусть скорость источника A - V_c букв в секунду.

Тогда можно считать, что на одну букву приходится H(A) битов. Тогда средняя битовая скорость источника $\nu_c = V_c H$(A). Реальная битовая скорость зависит от того, какие буквы передаются, и, конечно, может отличаться от средней.

Представим, что битовый поток поступает в канал (без искажений).



Проблема: синхронизация скорости источника и пропускной способности канала.

Очевидно, что если по каналу передавать информацию со скоростью $v_k < V_c \log n$, то могут произойти потери информации, так как $\max H(A) = \log n$. Если же технические возможности канала таковы, что он обеспечивает скорость $v_k \geq V_c \log n$, то информация может быть передана без потерь.

В общем случае с использованием неравенства Чебышева (закона больших чисел) можно показать, что справедлива следующая теорема.

Теорема. (Теорема Шеннона для канала без шума). Для источника A с энтропией $H(A)$ с вероятностью, стремящейся к единице при $n \to \infty$, скорость v передачи по каналу без шума удовлетворяет условиям:

1) При $v < Vc\ H(A)$ невозможна гарантированная передача без потерь.
2) Существует такая кодировка букв источника битовыми словами, что для любого $\varepsilon > 0$ и при $v > Vc\ H(X) + \varepsilon$ передача (с вышеприведенной вероятностью) возможна.

На самом деле задача сложнее, если учесть **необходимость** синхронизировать не только скорость передачи и пропускную способность канала, но еще и скорость кодирования символов.

7.4 Упражнения.

1. Пусть l_i – число слов длины i во множестве B слов некоторого префиксного кода. Доказать (не используя неравенство Крафта!) неравенство $\sum_i l_i 2^{-i} \leq 1$.

2. Проверить неравенство Крафта и построить префиксный код при q=3 с длинами слов: 1,1,3,4,4,4.

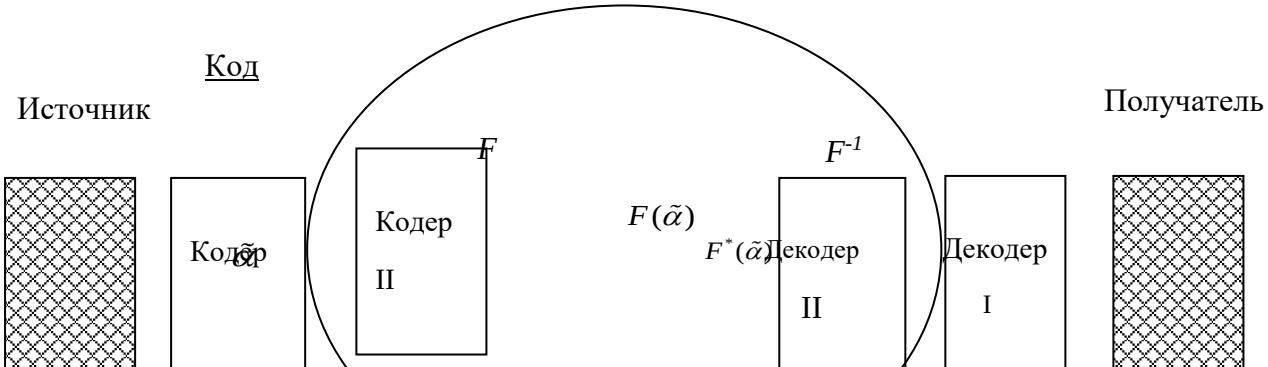
3. Построить два кода: Шеннона и Хаффмена. Вероятности букв: 0.2, 0.15, 0.15, 0.1, 0.1, 0.1, 0.06, 0.05, 0.05, 0.04. Найти стоимость кодирования и энтропию.

4. Вероятность 0 – 1/5, вероятность 1 - 4/5. Построить блочный код Хаффмена (до длины блока – 3) и арифметический код с 8 словами. Найти стоимость кодирования и энтропию.

8. Лекция 8. Передача информации с искажением. Выпадения символа из переданного слова. Проблема восстановления информации.

8.1 Передача информации по каналу с шумом.

Рассмотрим теперь следующую схему.



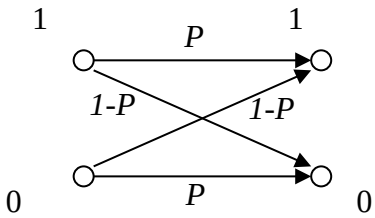
Потери информации

a_1 ... a_n — Избыточность — a_1 ... a_n

Часть рисунка вне овала соответствует изучаемой раньше схеме, т.е. кодер (*Кодер I*) и декодер (*Декодер I*) – это, например, устройства реализующие алгоритм Хаффмена или что-то подобное. Но в отличие от предыдущего случая (канала без шума) добавился еще один кодер (*Кодер II*) и декодер (*Декодер II*). Их задача – обеспечить защиту передаваемой информации от шума в канале. Под *шумом* можно понимать самые разные искажения сигнала при передаче по каналу. Примеры моделей этих искажений приведены в следующем разделе.

8.2 Модели каналов.

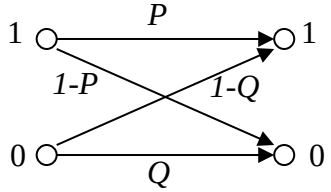
1) Двоичный симметричный канал



P > 1/2

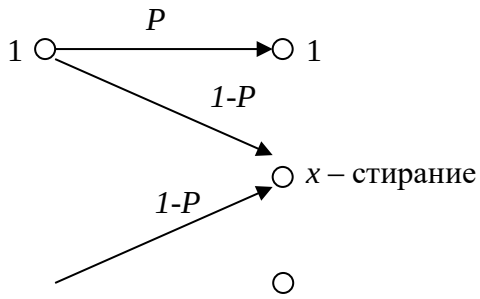
Канал называется двоичным, потому, что по нему передаются два сигнала: 0 и 1. Симметричность заключается в том, что они подвергаются искажениям в одинаковой степени. Вероятность правильной передачи P>1/2. (Ниже вероятность ошибки для удобства будем обозначать через p=1-P. При этом всегда будет понятно, о чем идет речь. Заметим, что критичный случай именно p=1/2. Если, например, p>1/2, то на приеме можно поменять местами 0 и 1. Ниже будет показано, что случай p=1/2 соответствует нулевой пропускной способности канала. Но это понятно и без всякой математики: что бы не попало в канала – на выходе с равной вероятностью будет принят или 0 или 1.

2) Двоичный ассиметричный канал



P > 1/2, Q > 1/2

3) Двоичный стирающий канал



Пример:

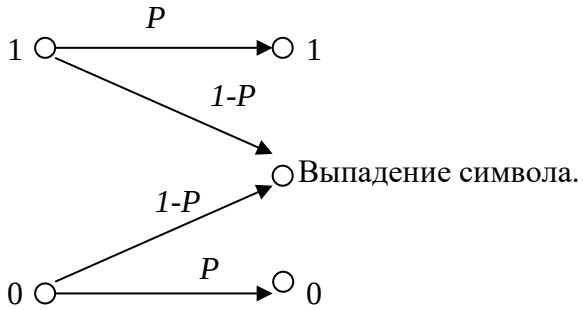
Передаем 11011, p = 4/5, ошибка в четвертом символе.

Для двоичного канала на приеме может быть получена последовательность : 11001

Для канала со стиранием в этом случае будем иметь : 110x1.

Это уже существенно другой тип искажений. В отличие от предыдущих случаев, здесь мы всегда уверены в том, что полученные 0 или 1 были именно такими и переданы, а на месте искаженных символов принимает новый символ *x*. В предыдущих случаях мы не знаем, какие из полученных сигналов искажены (и есть ли искажения вообще). Нахождение и исправление таких искажений приводит к сложным математическим задачам *Теории кодирования*. В случае же канала со стиранием такой подход тоже можно использовать, но можно свести задачу на уровень *Протокола*. Например, можно просто перезапрашивать искаженные символы.

4) Канал с выпадением.



Пример:

Передаем 11011, p = 4/5, ошибка в четвертом символе.

Для двоичного канала на приеме может быть получена последовательность: 11001

Для канала с выпадением в этом случае будем иметь: 1101.

8.3 Пример кода для канала с выпадением.

Построение кодов для таких каналов – сложные комбинаторные задачи. Проблема соотношения пропускной способности канала и скорости передачи кода отходит при этом на второй план. Приведем простейший пример.

Пусть дан код C, $\alpha=(\alpha_1,\alpha_2,\ldots,\alpha_n)$ – кодовое слово. Положим $w(\alpha)=\sum_{i=1}^{n}\alpha_i i$.

Опр. C_{nk} – кодом называется множество слов α из B^n таких, что $w(\alpha)=0 \pmod k$.

Утв. C_{nk} – код является кодом с исправлением одного выпадения при k>n.

Доказательство. Пусть в слове α выпал символ. В результате получилось слово $\beta=(\beta_1, \beta_2,\ldots, \beta_{n-1})$ длины n-1. Пусть правее выпавшего символа расположено N_1 единиц и N_0

нулей. Тогда, если выпал 0, то $w(\alpha)-w(\beta)= N_1$, а, если 1, то $w(\alpha)-w(\beta)= N -N_0$. В обоих случаях

$0 \le w(\alpha)-w(\beta) \le N < k$.

Пусть $\Delta\beta$ – наименьший неотрицательный вычет числа $w(\beta)$ по mod k (Это разность kp-$w(\beta)$, где kp – минимальное число, кратное k и не меньшее $w(\beta)$) .

Тогда из того, что $w(\alpha)=0 \pmod k$ получаем $w(\alpha)-w(\beta)= \Delta\beta$. Так как $N_1 \le |\beta| < N -N_0$, то на основании сравнения чисел $|\beta|$ и $\Delta\beta$ можно определить выпавший символ.

1. Если $|\beta| \ge \Delta\beta$, то нужно вставить в слово символ 0 так, чтобы правее него было $\Delta\beta$ единиц.
2. В противном случае вставляется 1 так, чтобы правее нее было n- $\Delta\beta$ нулей.

Утверждение доказано.

Пример.

Пусть N=6 и k=7. Если в слове α=110100 выпал первый символ, то получилось β=10100. В этом случае: $|\beta|$ =2, $w(\beta)$=4, поэтому $\Delta\beta$=3. Так как $|\beta| < \Delta\beta$, то исходное слово получается, если в принятом слове отсчитать справа n- $\Delta\beta$ нулей и поставить перед ними 1.

Упражнение. В качестве упражнения можете проверить, что по подобному принципу строятся коды, которые могут исправлять или одну ошибку типа вставки (вместо выпадения появляется лишний символ), или одну ошибку замещения 0 на 1.

Упражнение. При $k \ge 2n$ аналогичным образом можно построить, который исправляет одну любую из этих трех видов ошибок.

Если же выпадает несколько символов, то задача декодирования может сводится к задаче *восстановления слова по фрагментам*.

Зная n и p можно прогнозировать длины фрагментов l. Критерий восстановления слова по фрагментам с использованием этих параметров даст k – необходимое количество фрагментов для восстановления. Отсюда можно построить алгоритм декодирования с перезапросами: декодер, получив фрагмент принимает решение о его перезапросе. Набрав необходимое количество фрагментов, декодер произведет декодирование.

Заметим, что проколы такого типа (Ack – Nac) используются на втором уровне модели OSI для случая *кодов с обнаружением* ошибок. (В таких кодах декодер не производит исправления ошибки, а только фиксирует ее появление. В случае же этого самого появления и используется протокол с перезапросом.)

8.4 Упражнения.

1. Восстановить выпавший символ 10100101. (n=8, k=9).

2. Восстановить выпавший символ 1010100101.

3*. Используя описанный в предыдущем разделе подход, предложить алгоритм исправления одной вставки.

4*. Используя описанный в предыдущем разделе подход, предложить алгоритм исправления одной замены (0 на 1 или 1 на 0).

9. Лекции 9. Проблема восстановления слов.

Представим себе ситуацию, когда отправитель передает не слово целиком, а его части (подслова или фрагменты), а получателю нужно восстановить по ним все слово. Или схожая с описанной в предыдущем разделе ситуация, когда отправитель несколько раз передает одно и то же слова, а в канале происходит не одно выпадения, а несколько. В этом случае получатель вновь имеет только части слова.

9.1 Словарная модель анализа информации

Общая проблема, которую мы будем обсуждать в этом разделе, состоит в следующем: в какой мере слово определяется своими частями? Точная постановка задачи состоит в следующем.

Пусть $I_n = [1,2,\dots,n]$ — отрезок натурального ряда и v $=$ $(i_1 i_2 \dots i_k) \subseteq$ I_n. Тогда подмножество v «вырезает» из слова x $= x_1 x_2 \dots x_n$ фрагмент по следующему правилу:

$$x_v = x_{i_1} x_{i_2} \dots x_{i_k}.$$

Ясно, что $|x_v| = k = |v|$.

Если $x_v = G = G_1 G_2 \dots G_k$, то всё множество слов из B^n таких, что $x_v = G$, удобно описать следующим образом: $x_{i_1}^{G_1} x_{i_2}^{G_2} \dots x_{i_k}^{G_k}$. Таким образом, искомое множество слов — это множество решений следующего логического уравнения:

$$x_v^G = x_{i_1}^{G_1} x_{i_2}^{G_2} \dots x_{i_k}^{G_k} = 1. \quad (9.1)$$

В стандартных логических обозначениях x_v^G — это конъюнкция ранга k или $(n-k)$-мерная грань в B^n.

Пример. Если $v = (124) \subseteq I_4$ и $G = (010)$, то соответствующая конъюнкция имеет вид: $x_v^G = \overline{x_1} x_2 \overline{x_4}$. Множество «единиц» этой конъюнкции следующее: $y_1 = (0100)$, $y_2 = (0110)$.

В общем случае будем считать, что информацию о неизвестном слове $x = (x_1 x_2 \dots x_n)$ мы получаем с помощью характеристического множества $V = \{v_1 \dots v_N\}$, где $v_k \subseteq I_n$ для $k = \overline{1, N}$.

Определение. «Информацией» о слове $a \in B^n$ называется мультимножество фрагментов $S_a = \{a_{v_1} a_{v_2} \dots a_{v_N}\}$, где фрагменты a_{v_k} были определены выше.

Заданное характеристическое множество $V = \{v_1 \dots v_N\}$ порождает следующее отношение V-эквивалентности на множестве слов B^n.

Слова $a, b \in B^n$ называются V-эквивалентными, если выполняется соотношение

$$S_a = S_b, \quad (9.2)$$

которое означает, что мультимножества $\{a_{v_k}\}$ и $\{b_{v_k}\}$ совпадают. Это отношение мы будем обозначать следующим образом: $a \overset{V}{\sim} b$.

Примеры.

1) Если $V = \{v = (12 \dots n)\}$, то

$$a \overset{V}{\sim} b \rightleftarrows a = b.$$

В этом случае имеется 2^n классов эквивалентности.

2) Если $V = \{(1), (2), \dots, (n)\}$, то

$$a \overset{V}{\sim} b \rightleftarrows ||a|| = ||b||.$$

В этом случае имеется ровно $(n + 1)$ класс эквивалентности.

Таким образом, каждое характеристическое множество V порождает разбиение B^n на классы эквивалентности.

$$B^n = \bigcup_a V_a,$$

где V_a — класс эквивалентности, включающий в себя слово a. В содержательном смысле мощность $|V_a|$ класса эквивалентности V_a характеризует ту точность, с которой можно определить слово $a \in B^n$ с помощью характеристического множества V. Если $|V_a| = 1$ для всех $a \in B^n$, то V однозначно определяет любое слово $a \in B^n$ своими фрагментами. Каждое такое множество мы будем называть *полным*. Описание всех полных множеств представляет собой любопытную задачу, о решении которой речь будет идти ниже.

9.2 Алгоритм на основе логического перманента.

Пусть $f_{ij}(x_1 \dots x_m)$ — логические функции от m переменных и $A = ||f_{ij}||$, где $i, j = \overline{1, N}$.

Определение. Логическим перманентом матрицы $A = ||f_{ij}||$ называется следующая Д.Н.Ф.:

$$per\ A = \bigvee_{\{g\}} f_{1_{g_1}} f_{2_{g_2}} \dots f_{N_{g_N}} \qquad (9.3)$$

где $g \in S_N$ и S_N — симметрическая группа порядка N.

Таким образом, Д.Н.Ф. $per\ A$ содержит $N!$ конъюнкций. Как обычно, $N_{per\ A} = \{x: per\ A(x) = 1\}$ — это множество единиц логической функции $per\ A$.

Пример. Пусть $f_{11} = x_1\overline{x_2}x_3$, $f_{12} = \overline{x_1}x_2\overline{x_3}$, $f_{21} = x_2\overline{x_3}x_4$, $f_{22} = \overline{x_2}x_3\overline{x_4}$. Тогда

$$A = \begin{Vmatrix} x_1\overline{x_2}x_3 & \overline{x_1}x_2\overline{x_3} \\ x_2\overline{x_3}x_4 & \overline{x_2}x_3\overline{x_4} \end{Vmatrix}$$

и

$$per\ A = x_1\overline{x_2}x_3\overline{x_4} \vee \overline{x_1}x_2\overline{x_3}x_4$$

Далее

$$N_{per\ A} = \{1010, 0101\}.$$

Пусть, как и ранее, V_a — класс эквивалентности, построенный по характеристическому множеству V и содержащий слово $a \in B^n$.

Конструкция V_a описывается следующими шагами.

1) При заданном $V = \{v_1 \dots v_N\}$ и $a \in B^n$ находим слова

$$e_i = a_{v_i}, \qquad i = \overline{1,N}.$$

2) Строим конъюнкции

$$x_{v_i}^{a_{v_j}} = x_{v_i}^{e_j}, \qquad i,j = \overline{1,N}.$$

При этом, если $|a_{v_j}| \neq |v_i|$, то соответствующая конъюнкция — это тождественный нуль.

3) Строим матрицу $A = \left\| \begin{matrix} a_{v_j} \\ x_{v_i} \end{matrix} \right\|$.

4) $\mathrm{V_a} = N_{per\,A}$

Обоснование пункта 4) довольно просто и состоит в следующем. Любое слово $x \in V_a$ обладает следующим характеристическим свойством. Всё мультимножество фрагментов $\{x_{v_i}\}$ совпадает с мультимножеством $\{e_i\}$. Другими словами, для некоторой перестановки $g \in S_N$ справедливы соотношения

$$x_{v_k} = e_{g(k)}, k = \overline{1,N}.$$

Решению системы соответствует решение уравнения

$$x_{v_1}^{e_{i_1}} x_{v_2}^{e_{i_2}} \dots x_{v_N}^{e_{i_N}} = 1, \quad (9.4)$$

что соответствует элементу логического перманента матрицы A.

Пример. Пусть $n = 4$, $V = \{v_1 = (123), v_2 = (234)\}$, $a = 1010$.

Применим к этому случаю алгоритм построения класса эквивалентности V_a.

1) $e_1 = a_{v_1} = 101$, $e_2 = a_{v_2} = 010$.

2) $x_{v_1}^{e_1} = x_1\overline{x_2}x_3$, $x_{v_1}^{e_2} = \overline{x_1}x_2\overline{x_3}$, $x_{v_2}^{e_1} = x_2\overline{x_3}x_4$, $x_{v_2}^{e_2} = \overline{x_2}x_3\overline{x_4}$.

3) $A = \left\| \begin{matrix} x_1\overline{x_2}x_3 & \overline{x_1}x_2\overline{x_3} \\ x_2\overline{x_3}x_4 & \overline{x_2}x_3\overline{x_4} \end{matrix} \right\|$,

$$per\ \mathrm{A} = x_1\overline{x_2}x_3\overline{x_4} \vee \overline{x_1}x_2\overline{x_3}x_4$$

и $V_a = \{1010, 0101\}$.

В общем виде матрица A имеет следующие параметры. Если $\mathrm{V} = \{\mathrm{v_1 v_2} \dots \mathrm{v_N}\}$, то порядок A равен N; при $|v_k| = r$ ранг конъюнкции $x_{v_k}^{e_i}$ равен r. Далее: k-я строка матрицы A имеет вид: $\left\| x_{v_k}^{e_1} x_{v_k}^{e_2} \dots x_{v_k}^{e_N} \right\|$, а k-ый столбец

$$\left\| \begin{matrix} x_{v_1}^{e_k} \\ x_{v_2}^{e_k} \\ \dots \\ x_{v_N}^{e_k} \end{matrix} \right\|.$$

Пример. Если $n = 3$, $V = \{v_1 = (12), v_2 = (13), v_3 = (23)\}$ и $a = 011$,

то $e_1 = 01$, $e_2 = 01$, $e_3 = 11$. Отсюда следует

$$A = \begin{Vmatrix} \overline{x_1}x_2 & \overline{x_1}x_2 & x_1x_2 \\ \overline{x_1}x_3 & \overline{x_1}x_3 & x_1x_3 \\ \overline{x_2}x_3 & \overline{x_2}x_3 & x_2x_3 \end{Vmatrix}.$$

Разлагая $per\ A$ по первой строке, получаем выражение:

$$per\ A = \overline{x_1}x_2\ per \begin{Vmatrix} \overline{x_1}x_3 & x_1x_3 \\ \overline{x_2}x_3 & x_2x_3 \end{Vmatrix} \vee\ \overline{x_1}x_2\ per \begin{Vmatrix} \overline{x_1}x_3 & x_1x_3 \\ \overline{x_2}x_3 & x_2x_3 \end{Vmatrix} \vee\ x_1x_2\ per \begin{Vmatrix} \overline{x_1}x_3 & \overline{x_1}x_3 \\ \overline{x_2}x_3 & \overline{x_2}x_3 \end{Vmatrix} =$$

$$= \overline{x_1}x_2(\overline{x_1}x_2x_3 \vee x_1\overline{x_2}x_3) \vee \overline{x_1}x_2(\overline{x_1}x_2x_3 \vee x_1\overline{x_2}x_3) \vee x_1x_2(\overline{x_1x_2}x_3 \vee \overline{x_1}x_2x_3) =$$

$$= \overline{x_1}x_2x_3$$

и, таким образом, $V_a = \{a = (011)\}$.

В общем виде Д.Н.Ф. $per\ A$ имеет $(N!)$ конъюнкций, максимальный ранг которых не превосходит $\sum_{i=1}^{N} |\,v_i|$ с учётом кратности вхождения букв и не превосходит n при классическом определении. Следует отметить также, что мы имеем дело не с «элементарными» конъюнкциями, а с некоторым их обобщением, представляющим специальный класс булевых функций, включающий в себя элементарные конъюнкции.

Следствие. $|V_a| = \|per\ A_a\|$, где под нормой $per\ A_a$, понимается число единиц логической функции $per\ A_a$.

Следующий частный случай служит хорошей иллюстрацией для всех приведённых выше абстрактных построений.

Пусть $V = \{(i,j), 1 \le i < j \le n\}$ — характеристическое множество, которое соответствует всем парам букв x_ix_j неизвестного слова $x = (x_1 \dots x_n)$. Если $a = (\alpha_1\alpha_2 \dots \alpha_n)$, то класс эквивалентности V_a выглядит следующим образом — это логический перманент матрицы A:

$$A = \begin{Vmatrix} x_1^{\alpha_1}x_2^{\alpha_2} & x_1^{\alpha_1}x_2^{\alpha_3} & \dots & x_1^{\alpha_{n-1}}x_2^{\alpha_n} \\ x_1^{\alpha_1}x_3^{\alpha_2} & x_1^{\alpha_1}x_3^{\alpha_3} & \dots & x_1^{\alpha_{n-1}}x_3^{\alpha_n} \\ \dots & \dots & \dots & \dots \\ x_{n-1}^{\alpha_1}x_n^{\alpha_2} & x_{n-1}^{\alpha_1}x_n^{\alpha_3} & \dots & x_{n-1}^{\alpha_{n-1}}x_n^{\alpha_n} \end{Vmatrix}.$$

В частности, для $n = 4$, $a = (1001)$ матрица A имеет вид

$$A = \begin{Vmatrix} x_1\overline{x_2} & x_1\overline{x_2} & x_1x_2 & \overline{x_1x_2} & \overline{x_1}x_2 & \overline{x_1}x_2 \\ x_1\overline{x_3} & x_1\overline{x_3} & x_1x_3 & \overline{x_1x_3} & \overline{x_1}x_3 & \overline{x_1}x_3 \\ x_1\overline{x_4} & x_1\overline{x_4} & x_1x_4 & \overline{x_1x_4} & \overline{x_1}x_4 & \overline{x_1}x_4 \\ x_2\overline{x_3} & x_2\overline{x_3} & x_2x_3 & \overline{x_2x_3} & \overline{x_2}x_3 & \overline{x_2}x_3 \\ x_2\overline{x_4} & x_2\overline{x_4} & x_2x_4 & \overline{x_2x_4} & \overline{x_2}x_4 & \overline{x_2}x_4 \\ x_3\overline{x_4} & x_3\overline{x_4} & x_3x_4 & \overline{x_3x_4} & \overline{x_3}x_4 & \overline{x_3}x_4 \end{Vmatrix}.$$

и

$$per\ \mathrm{A} = \ x_1\overline{x_2x_3}x_4 \vee \overline{x_1}x_2x_3\overline{x_4}.$$

Конъюнкция $x_1\overline{x_2x_3}x_4$ соответствует главной диагонали матрицы A, а конъюнкция $\overline{x_1}x_2x_3\overline{x_4}$ — второй главной диагонали этой матрицы. Ясно, что $V_a = \{(1001), (0110)\}$.

9.3 Пример на основе k-го слоя B^n.

Одним из самых простых и естественных характеристических множеств является k-ый слой B^n или множество B_k^n

$$B_k^n = \{x: ||x|| = k\} = \{i_1 i_2 \dots i_k, \text{где } 1 \le i_1 < i_2 < \dots < i_k \le n\}.$$

В содержательном смысле условие $V = B_k^n$ означает, что информацией о неизвестном слове $x \in B^n$ являются все его фрагменты длины k.

1) Пусть $k = 1$.

В этом случае

$$V_a = \{1^{||a||}, 0^{n-||a||}\}$$

и каждый класс эквивалентности представляет собой слой куба B^n. Таким образом, имеется ровно $(n+1)$ класс эквивалентности.

2) $k = 2$.

В этом случае $V = B_2^n$ и критерий эквивалентности двух слов $x = (x_1 \dots x_n)$ и $y = (y_1 \dots y_n)$ состоит в выполнении двух условий:

$$\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i \,, \sum_{i=1}^{n} i\, x_i = \sum_{i=1}^{n} i\, y_i. \quad (9.5)$$

Условия (9.5) означают, что эквивалентные слова имеют одинаковый вес Хэмминга и одинаковую сумму номеров единичных координат. Для нахождения числа классов эквивалентности используем следующее утверждение.

Теперь заметим, что

$$\min_x f = \sum_{i=1}^{k} i = \frac{k(k+1)}{2}, \qquad \max_x f = \sum_{i=n-k+1}^{n} i = \frac{k(2n-k+1)}{2}.$$

Предложение. Число классов эквивалентности для характеристического множества $V = B_2^n$ равно $\frac{(n+1)(n^2-n+6)}{6}$.

С учётом предыдущих замечаний доказательство этого предложения состоит в нахождении разности $\max f - \min f$. Доказательство этого утверждения выносится в качестве упражнения. Это задача из задания.

Пример. Для $n = 4$ разбиение на кассы эквивалентности имеет следующий вид:

$$k = 0: \{(0000)\};$$
$$k = 1: \{(1000)\}, \{(0100)\}, \{(0010)\}, \{(0001)\};$$
$$k = 2: \{(1100)\}, \{(1010)\}, \{(0110)\}, \{(1001)\}, \{(0101)\}, \{(0011)\};$$
$$k = 3: \{(1110)\}, \{(1101)\}, \{(1011)\}, \{(0111)\};$$
$$k = 4: \{(1111)\}.$$

Общее число классов эквивалентности равно 15.

Пример. Для $n = 5$:

$k = 0: \{(00000)\}, \quad 1;$
$k = 1: \{(10000)\}, \{(01000)\}, \{(00100)\}, \{(00010)\}, \{(00001)\}, \quad 5;$
$k = 2: \{(\overset{3}{11000})\}, \{(\overset{4}{10100})\}, \{(\overset{5}{10010})\}, \{(\overset{6}{10001})\}, \{(\overset{5}{01100})\}, \{(\overset{6}{01010})\},$
$\{(\overset{7}{01001})\}, \{(\overset{7}{00110})\}, \{(\overset{8}{00101})\}, \{(\overset{9}{00011})\}, \{\dot{3}, \dot{4}, 5,6,7, \dot{8}, \dot{9}\} = 7;$
$k = 3: \{(\overset{6}{11100})\}, \{(\overset{7}{11010})\}, \{(\overset{8}{11001})\}, \{(\overset{8}{10110})\}, \{(\overset{9}{10101})\}, \{(\overset{10}{10011})\},$
$\{(\overset{9}{01110})\}, \{(\overset{10}{01101})\}, \{(\overset{11}{01011})\}, \{(\overset{12}{00111})\}, \{\dot{6}, \dot{7}, 8,9,10, \dot{11}, \dot{12}\} = 7;$
$k = 4: \{(\overset{10}{11110})\}, \{(\overset{11}{11101})\}, \{(\overset{12}{11011})\}, \{(\overset{13}{10111})\}, \{(\overset{14}{01111})\},$
$\{10,11,12,13,14\} = 5;$
$k = 5: \{(11111)\} = 1.$

Общее число классов эквивалентности 26. Классы эквивалентности следующие: (00000). (10000), (01000), (00100), (00010), (00001). (11000), (10100), (10010), (10001), (01001), (00101), (00011). (11100), (11010), (11001), (10101), (10011), (01011), (00111). (11110), (11101), (11011), (01111), (10111). (11111).

Для $V = B_3^n$ биномиальные коэффициенты $\binom{x}{\alpha\beta\gamma}$ вычисляются следующим образом.

По определению

$$\binom{x}{110} = \sum_{i<j<k} x_i\, x_j (1 - x_k),$$
$$\binom{x}{101} = \sum_{i<j<k} x_i\, (1 - x_j) x_k,$$
$$\binom{x}{011} = \sum_{i<j<k} (1 - x_i) x_j x_k.$$

Так как

$$\binom{x}{a} = \binom{\overline{x}}{\overline{a}},$$

то приведённых выше формул достаточно для нахождения всех остальных биномиальных коэффициентов: $\binom{x}{100}, \binom{x}{010}, \binom{x}{001}$.

Далее

$$\binom{x}{111} = \binom{||x||}{3}, \binom{x}{000} = \binom{n - ||x||}{111}.$$

Подробные рутинные выкладки приводят к следующим полиномам, устанавливающим эквивалентность слов по характеристическому множеству B_3^n.

Пусть

$$\Psi_1(x) = \sum_{i=1}^{n} x_i\,, \Psi_2(x) = \sum_{i=1}^{n} i\, x_i, \Psi_3(x) = \sum_{i=1}^{n} i^2\, x_i,$$

$$\Psi_4(x) = \sum i < j i x_i x_j, \Psi_5(x) = \sum_{i<j} j\, x_i x_j.$$

Если перейти в другую систему координат $(y_1 y_2 \ldots y_r)$, где y_i — это номер i-й единичной координаты в слове $x = (x_1 x_2 \ldots x_n)$, то приведённые выше функции перейдут в следующие:

$$\Psi_1^0(x) = r, \Psi_2^0(x) = \sum_{i=1}^{r} y_i\,, \Psi_3^0(x) = \sum_{i=1}^{r} i\, y_i, \Psi_4^0(x) = \sum_{i=1}^{r} y_i^2.$$

Предложение. Справедливо соотношение

$$x \overset{V}{\sim} y \leftrightarrows \Psi_i^0(x) = \Psi_i^0(y), i = \overline{1,4}. \quad (9.6)$$

Таким образом, полиномы $\Psi_1^0, \Psi_2^0, \Psi_3^0, \Psi_4^0$ представляют собой полную систему инвариантов любого класса эквивалентности по характеристическому множеству B_3^n. Если при этом учесть число значений, которое принимает каждая из функций $\Psi_i^0(x)$, а это, по порядку величины не превосходящие n, n^2, n^3, n^3, то можно получить оценку сверху для числа классов эквивалентности: $O(n^9)$.

В общем виде система инвариантов строится следующим образом.

В соответствии со значениями биномиальных коэффициентов

$$\binom{x_1 x_2 \dots x_n}{\alpha_1 \alpha_2 \dots \alpha_k}$$

выписываются следующие полиномы:

$$\sum_{1 \le i_1 < i_2 < \dots < i_k \le n} x_{i_r}, r = \overline{1,k},$$

$$\sum_{1 \le i_1 < i_2 < \dots < i_k \le n} x_{i_r} x_{i_s}, r, s = \overline{1,k},$$

...

и так далее вплоть до

$$G_k = \sum_{1 \le i_1 < i_2 < \dots < i_k \le n} x_{i_1} x_{i_2} \dots x_{i_k}. \quad (9.7)$$

Всего полиномов вида (9.7) существует ровно $\binom{k}{1} + \binom{k}{2} + \dots = 2^k - 1$ штук.

9.4 Упражнения.

1. Найти V_a, если n=4, a=0101, V= {v_1=(123), v_2=(234)}.

2. Найти V_a, если n=4, a=1101, V= {v_1=(124), v_2=(134)}.

3*. Доказать, что число классов эквивалентности для характеристического множества $V = B_2^n$ равно $\frac{(n+1)(n^2-n+6)}{6}$.

10. Лекция 10. Распознавание слов по подсловам и фрагментам.

10.1 Распознавание слов по подсловам.

Если характеристическое множество V_k имеет вид

$$V_k = \{v_1 = (1,2,\dots,k), v_2 = (1,2,\dots,k-1,k+1), \dots,$$
$$v_{n-k+1} = (n-k+1, n-k+2, \dots, n)\},$$

то содержательно это означает, что в качестве «частей» неизвестного слова x рассматриваются его подслова длины k.

Пусть

$$V_{\le k} = \bigcup_{i=1}^{k} V_i.$$

Ясно, что это характеристическое множество содержит ровно $r = \frac{k(2n-k+1)}{2}$ элементов и, таким образом, информация о неизвестном слове x — это r слов разных длин: от 1 до k. Проблема состоит в следующем: какой длины k достаточно для однозначного восстановления любого бинарного слова длины n по фрагментам, порождаемым характеристическим множеством $V_{\leq k}$.

Рассмотрим следующие слова:

$$a_m = 1^{m-1}01^{m-2}, b_m = 1^{m-2}01^{m-1}.$$

Утверждение. Справедливо соотношение: $a_m \overset{V_{\leq k}}{\sim} b_m$ для $k \leq m-2$.

Доказательство. Непосредственное вычисление показывает, что

$$\binom{a_m}{1^r} = \binom{b_m}{1^r},\ r = 1,2,\dots,m-2.$$

Подслова a_m и b_m, имеющие две и три серии, выглядят следующим образом: 1^r0, 01^r, 1^p01^q. При $k \leq m-2$ такие слова входят в a_m и b_m с одинаковой кратностью.

Из этого утверждения следует, что при длине слова 2m+2 подслов длины, не большей m-2 не достаточно, для восстановления слова.

Следующая теорема показывает, что длины m+2 уже хватает.

Теорема 10.1. Если $k \geq \frac{n}{2} + 1$, то любое слово $x \in B^n$ однозначно определяется подсловами длины $\leq k$.

Схема доказательство. Пусть $x = (x_1 \dots x_n)$. Выпишем все подслова слова x длины $k, k-1, \dots, 1$ в отдельные таблицы.

$x_1 \dots x_k$	$x_1 \dots x_{k-1}$	...	x_1
$x_2 \dots x_{k+1}$	$x_2 \dots x_k$	...	x_2
...	...	...	...
$x_{n-k+1} \dots x_n$	$x_{n-k+2} \dots x_n$	...	x_n

Возьмём произвольную таблицу из выписанных и найдём сумму элементов по всем столбцам этой таблицы. Так как эти суммы не зависят от расположения строк в таблице, то мы получаем систему инвариантов, которые можем использовать для составления системы линейных уравнений, правыми частями которых и служат вычисленные выше суммы.

Для пояснения этой процедуры рассмотрим случай $n = 4$ и $k = 3$.

		x_1
	x_1x_2	x_2
$x_1x_2x_3$	x_2x_3	x_3
$x_2x_3x_4$	x_3x_4	x_4
$t_6t_5t_4$,	t_3t_2,	t_1

(10.1)

Из таблиц (10.1) получаем систему уравнений

$$\begin{matrix} x_1+x_2+x_3+x_4=t_1 \\ x_2+x_3+x_4=t_2 \\ x_3+x_4=t_4 \end{matrix} \quad , \quad \begin{matrix} x_1+x_2=t_6 \\ x_1+x_2+x_3=t_3 \\ x_2+x_3=t_5 \end{matrix}$$

Решив эту систему, получаем значения неизвестных x_i:

$$x_1=t_1-t_2, x_2=t_2-t_4, x_3=t_5-t_2+t_4, x_4=t_2-t_5.$$

10.2Распознавание слов по фрагментам

В целом ряде случаев параметры $\binom{x}{a}$ позволяют однозначно определить слово x или найти какие-то ограничения на неизвестное слово.

Пусть $Y_1, Y_2, \dots, Y_k$ — номера единиц в двоичном слове $x \in B_k^n$.

Лемма. Пусть $e_r = 1^{r-1}01^{k-r+1},\ r = 1,2,\dots,k$. Тогда

$$Y_1=\binom{x}{e_1}+1, Y_2=Y_1+\binom{x}{e_2}+1, \dots, Y_k=Y_{k-1}+\binom{x}{e_k}+1.$$

Доказательство. Так как, по определению,

$$x = 0^{Y_1-1}10^{Y_2}1\dots,$$

то $Y_1-1=\binom{x}{01^k}$, $Y_2=Y_1+\binom{x}{101^{k-1}}+1$ и так далее.

Пример.

Если $x=0^2 1^2 0^2 10$, то есть $||x||=3$, то

$$Y_1-1=\binom{x}{01^3}=2, Y_1=3,$$

$$Y_2-Y_1-1=\binom{x}{101^2}=0, Y_2=4,$$

$$Y_3-Y_2-1=\binom{x}{1^2 01}=2, Y_3=7.$$

Простым следствием приведённой выше леммы является следующее утверждение.

Теорема 10.2. Если $k > \left[\frac{n}{2}\right]$, то любое слово $x \in B^n$ однозначно определяется своими фрагментами длины $\le k$.

Доказательство. Так как

$$||x|| + ||\overline{x}|| = n,$$

то хотя бы одно из этих чисел не превосходит $\left[\frac{n}{2}\right]+1$. Вес $||x||$ слова x легко находится по фрагментам длины 1 и, если $||x|| \le \left[\frac{n}{2}\right]+1$, то с помощью предыдущей леммы мы восстанаваливаем слово x. Если же $||x|| > \left[\frac{n}{2}\right]+1$, то мы переходим к слову $\overline{x}$, заменяя каждый фрагмент его логическим отрицанием и, с помощью той же леммы, восстанавливаем слово $\overline{x}$ и, значит, слово x.

Теорема доказана.

В общем виде получение информации о координатах неизвестного слова $x = (x_1 x_2 \dots x_n)$ по характеристическому множеству $V = \{(i_1 i_2 \dots i_k), 1 \le i_1 < i_2 < \dots < i_k \le n\}$ выглядит следующим образом. Выпишем все фрагменты длины k слова x в матрицу A_x

$$A_x = \left\| \begin{matrix} x_1 & x_2 & \dots & x_k \\ x_1 & x_2 & \dots & x_{k+1} \\ \dots & \dots & \dots & \dots \end{matrix} \right\|.$$

Матрица $A_x = ||x_{ij}||$ имеет размеры $\binom{n}{k} \times n$. Пусть t_{ij} — кратность вхождения буквы x_j в j-ый столбец матрицы A. Прямое вычисление показывает справедливость формулы для t_{ij}.

Лемма. $t_{ij} = \binom{i-1}{j-1}\binom{n-i}{k-j}, i = \overline{1,n},\ j = \overline{1,k}$ (10.2)

Формула (10.2) показывает, что выполняется следующая система линейных уравнений:

$$\sum_{i=j}^{n-k+j} \binom{i-1}{j-1}\binom{n-i}{k-j} x_i = t_j, \qquad j = 1,2,\dots,k.$$

Здесь t_j — сумма элементов j-го столбца матрицы A_x.

10.3Дополнительная информация влечет дополнительную неопределенность.

Обратимся опять к исходной модели распознавания слов по фрагментам при заданном характеристическом множестве $V = \{v_1 \dots v_N\}$. Информация, полученная о слове $x = (x_1 \dots x_n)$ при помощи множества V, — это мультимножество фрагментов $\{x_{v_1}, x_{v_2}, \dots, x_{v_N}\}$. Если мы «расширим» характеристическое множество, то есть рассмотрим множество V_1 такое, что $V \subseteq V_1$, то информацией о слове x будет являться мультимножество фрагментов, содержащее исходное мультимножество $\{x_{v_1}, x_{v_2}, \dots, x_{v_N}\}$. Что при этом произойдёт с классами эквивалентности, порождёнными множеством V? Интуитивно очевидный ответ, состоящий в том, что классы эквивалентности не «расширятся», оказывается неверным. Рассмотрим следующий пример.

Пусть $n = 4$ и $V = \{v_1 = (14), v_2 = (23), v_3 = (24)\}$. Для любого слова $x = (x_1 x_2 x_3 x_4)$ имеем

$$x_{v_1} = x_1 x_4,$$

$$x_{v_2} = x_2 x_3,$$

$$x_{v_3} = x_2 x_4.$$

Отсюда получаем:

$$\begin{cases} x_1 + 2x_2 = t_1, \\ x_3 + 2x_4 = t_2. \end{cases} \qquad (10.3)$$

При любых натуральных t_1 и t_2 система (10.3) имеет единственное $(0,1)$ решение. Отсюда следует, что характеристическое множество V является полным, то есть каждый класс эквивалентности содержит ровно одно слово.

Рассмотрим теперь «расширенное» характеристическое множество $V_1 = V \cup (12)$. Если $a = (1001)$, то класс эквивалентности V_a определяется таким мультимножеством фрагментов: $\{a_v\} = \{a_{v_1} = (11), a_{v_2} = (00), a_{v_3} = (01), a_{12} = (10)\}$. Далее, если $b = (0110)$, то $\{b_v\} = \{b_{v_1} = (00), b_{v_2} = (11), b_{v_3} = (10), b_{12} = (01)\}$ и потому $a \overset{V_1}{\sim} b$. Таким образом, класс эквивалентности V_a содержит более одного элемента и множество V_1 не является полным.

Объяснить эту ситуацию можно следующим образом. Каждый фрагмент слова x несёт определённую информацию об этом слове. Но с этим же фрагментом связана и неопределённость, которая заключается в том, что неизвестно от «действия» каких элементов характеристического множества получился этот фрагмент. Потому «увеличивая» мощность характеристического множества мы, с одной стороны, поднимаем уровень информированности о неизвестном слове, а с другой стороны повышаем степень неопределённости о способе получения этой информации. От исхода такого рода коллизий и возникают эффекты типа отмеченного выше.

10.4 Полные характеристические множества.

Возвращаясь ко всему сказанному выше напомним, что информацию о неизвестном слове x мы получали в виде фрагментов этого слова, которые, в свою очередь, формировались с помощью характеристического множества $V = \{v_1 v_2 \dots v_N\}$. Возникает естественный вопрос о «структурных» свойствах множества V, которые влияют на количество информации, получаемой о неизвестном слове.

Определение. Характеристическое множество V называется *полным*, если $|V_a| = 1$ для любого слова $a \in B^n$.

Тривиальным примером полного характеристического множества является $V = \{v = (1,2,\dots,n)\}$. Если $V = \{v_1 = (1,2,\dots,k), v_2 = (k+1,\dots,2k+1)\}$, то множество V также является полным при $n = 2k+1$. Этот пример может быть обобщён следующим образом.

Если представление

$$[1,n] = \bigcup_{i=1}^{k} M_i$$

есть разбиение и $|M_i| \neq |M_j|$ для $i \neq j$, то характеристическое множество $V = \{M_1, M_2, \dots, M_k\}$ является полным.

Пример. Рассмотрим следующее характеристическое множество $\mathrm{V} = \{(2), (14), (3567)\}$. Если $x = (0110010)$, то фрагменты x следующие: $(1), (00), (1010)$. Восстановление x по этому множеству фрагментов производится очевидным образом: $\mathrm{x}_2 = 1, (\mathrm{x}_1\mathrm{x}_4) = (00), \ (\mathrm{x}_3\mathrm{x}_5\mathrm{x}_6\mathrm{x}_7) = (1010)$.

Отметим теперь, что множество $V = \{(123), (456)\}$ при $n = 6$ не является полным, так как

$$(111010) \overset{V}{\sim} (010111)$$

и, в общей форме, множество $V = \{v_1 = (1,2,\dots,k), (k+1,\dots,2k)\}$ также не является полным. Таким образом, разница между полными и неполными характеристическими множествами может быть незначительной.

Если $V_k = \{(i_1 \dots i_k), 1 \le i_1 < i_2 < \dots < i_k \le n\}$, то следующая конструкция даёт нижнюю границу для k, при котором V_k является полным множеством. Эту границу мы обозначим через $l(k)$.

Теорема 10.3. Справедливо соотношение

$$l(n) \ge \log_2 n.$$

Доказательство этого неравенства – сложная задача и не входит в нашу программу. Ниже приводятся некоторые соображения на эту тему и схема доказательства.

Схема доказательства.

Заметим, что слова (01) и (10) «неразличимы» по фрагментам длины 1 или формально

$$01 \overset{V_1}{\sim} 10.$$

Далее

$$0110 \overset{V_2}{\sim} 1001.$$

Действительно, по критерию (9.5)

$$\sum i x_i = 2 + 3 = \sum i y_i = 1 + 4.$$

Итерируя эту процедуру, получаем

$$a_k b_k \overset{V_{k-1}}{\sim} b_k a_k.$$

Так как длина слова $|a_k b_k| = 2^k$, то из следует, что

$$l(n) \ge \log_2 n.$$

10.5 Дополнения и замечания.

Отметим также, что каждое характеристическое множество V порождает разбиение B^n

$$B^n = \cup V_a$$

и энтропия такого разбиения равна следующей величине

$$H(V) = \sum_a |V_a| \log |V_a|.$$

Мощность каждого класса эквивалентности $|V_a|$ может быть найдена как число единиц логического перманента булевой матрицы, построенной по характеристическому множеству V. Такого рода вычисления были приведены выше.

Рассмотрим теперь следующую задачу. Дано некоторое множество слов $y_1 y_2 \dots y_m$, каждое из которых имеет длину k, то есть $|y_i| = k$. Существует ли слово y длины n такое, что $y_i < y,\ i = \overline{1, m}$.

Другими словами, существует ли слово y длины n такое, что каждое из слов y_i является фрагментом y? Таким образом, в этой задаче три параметра k, m, n и ответ, очевидно, зависит от их соотношений.

1) Если $2k \le n$, то ответ положительный для любого m. Действительно, слово $y = 1010 \dots 10 = (10)^k$ содержит в качестве фрагмента любое слово длины k.

2) Если $2k > n$, то ответ в общем виде негативный уже для $m = 2$, так как при $y_1 = 0^k$, $y_2 = 1^k$ никакое слово длины $< 2k$ не содержит y_1 и y_2 в качестве фрагментов.

Таким образом, неопределённым остаётся случай $2k \geq n$ и $\{y_1 \dots y_m\}$ — некоторое произвольное множество из B^{2k}. То есть для некоторого множества слов $y_1 y_2 \dots y_m$, каждое из которых имеет длину k, описать критерий существования слова y длины n такого, что $y_i < y,\ i = \overline{1,m}$.

Довольно просто решается также следующая задача распознавания: даны два слова a и x, является ли a фрагментом x? Если $a = \alpha_1 \alpha_2 \dots \alpha_m$, $x = x_1 x_2 \dots x_n$, то решение этой задачи выглядит следующим образом.

1) Выбираем число r_1, исходя из условия

$$r_1 = \min_i \{x_i = \alpha_1\}.$$

Если r_1 не существует, то ответ в исходной задаче: нет. Если r_1 существует, то переходим к шагу 2).

2) Рассмотрим слова $a_1 = \alpha_2 \dots \alpha_m$, $y_1 = x_{r+1} \dots x_n$ как новые входные данные и переходим к шагу 1).

Имеется довольно тесная и естественная связь между характеристическими множествами V и булевыми полиномами. Действительно, каждому булеву полиному

$$g(x) = \sum_{w \in W^0} x^w$$

можно сопоставить характеристическое множество V по следующему правилу: если $x^w = x_{i_1} x_{i_2} \dots x_{i_k}$, то $v = (i_1 \dots i_k) \in V$.

Обратное соответствие осуществляется очевидным способом: если $v = (i_1 \dots i_k) \in V$, то $x_v = x_{i_1} x_{i_2} \dots x_{i_k}$ является мономом полинома $g_V(x)$.

В терминах понятия V-эквивалентности эта связь имеет следующее выражение.

Предложение. Если $a \overset{V}{\sim} b$, то $g_V(a) = g_V(b)$.

Доказательство. По определению

$$g_V(a) = \sum_{v \in V} a^v.$$

По условию V-эквивалентности мультимножество $\{a_v\}$ однозначно определяется мультимножеством $\{b_v\}$, что и приводит к равенству $g_V(a) = g_V(b)$.

10.6 Упражнения.

1. Используя теорему о количестве вхождений одного слова в другое как фрагмента перевести сериальный код $0^3 1^2 010$ в последовательность номеров единичных позиций.

2**. Превратить схему доказательства теоремы 10.2 в строгое доказательство.

3***. Превратить схему доказательства теоремы 10.3 в строгое доказательство.

11. Лекция 11. Экстремальные задачи на метрических пространствах.

11.1Основные теоремы.

Дан булев куб размерности n с расстояние Хэмминга. В нем нужно разместить M точек, попарные расстояния между которыми не меньше. Пусть A(n,d) это максимальное значение M при таком размещении. Задача состоит в нахождении A(n,d) и оценок для него.

Лемма. Пусть есть множество $V=(\alpha_1 \dots \alpha_k)$, $V \subseteq B^n$, $d(V) \geq d$. Также есть множество $V'=(\beta_1 \dots \beta_S)$, $V' \subseteq B^n$ и для любых $x,y \in V'$: $\rho(x,y) \leq d-1$. Тогда множества $V \oplus \beta_1, V \oplus \beta_2, \dots, V \oplus \beta_S$ не пересекаются.

Доказательство. От противного. Если два множества пересекаются, то существуют четыре вектора, для которых справедливо соотношение:

$$\alpha_i \oplus \beta_j = \alpha_p \oplus \beta_r \Rightarrow \alpha_i \oplus \alpha_p = \beta_j \oplus \beta_r \Rightarrow |\alpha_i \oplus \alpha_p| = |\beta_j \oplus \beta_r|$$

где $|\alpha_i \oplus \alpha_p| \geq d$, $|\beta_j \oplus \beta_r| \leq d-1$. Получило противоречие.

Лемма доказана.

Теорема. (Граница Хемминга) Справедливо соотношение:

$$A(n,2t+1) \leq \frac{2^n}{\sum_{i=0}^{t} C_n^i}$$

Доказательство.

Пусть $V=(\alpha_1 \dots \alpha_k)$ – наш код, $V' = S_t(x) = \{y \in B^n : \rho(x,y) \leq t\}$ – шар радиуса t, тогда в этом шаре расстояние между двумя любыми точками не превосходит $\rho \leq 2t \leq d-1$.

Из леммы следует, что множества вида $V \oplus y$, где $y \in V'$, не пересекаются, но тогда

$$|V| \cdot |S_t(0)| \leq 2^n.$$

По определению $A(n,2t+1)$ и получим искомое неравенство.

Теорема доказана.

Теорема (Граница Джоши). Справедливо соотношение:

$$A(n,d) \leq 2^{n-d+1}.$$

Доказательство. Пусть V – множество векторов с попарным расстоянием не меньше расстоянием d, $V' \subseteq B^n$, V' – множество всех векторов y, у которых первые d-1

компонент произвольные, а остальные нули, т.е $y=(\forall \overset{\{0,1\}}{\underset{d-1}{символы}}\ 0...0)$. Тогда $|V'|=2^{d-1}$, для любых $x,y \in V'$: $\rho(x,y) \le d-1$.

Снова из леммы получаем

$$|V| \cdot 2^{d-1} \le 2^n.$$

Теорема доказана.

Теорема (Варшамова-Гильберта). Справедливо соотношение:

$$A(n, 2t+1) \ge \frac{2^n}{\sum_{i=1}^{2t} C_n^i}.$$

Доказательство. Рассмотрим простую геометрическую аналогию. Пусть дан отрезок [a,b]. Мы произвольным образом поочередно размещаем на нем отрезки длины d<(b-a) так, чтобы они не пересекались. Пусть мы разместили k таких отрезков, а (k+1)-й уже не помещается. После этого мы, оставляя на месте центры размещенных отрезков, увеличиваем их вдвое на одинаковую величину в обе стороны. После этой операции на отрезке не останется пустого места, ведь, если бы оно было, то от любой точки на этом пустом месте, до центра одного из соседних размещенных отрезков было бы расстояние, большее 2d, а это противоречит тому, что (k+1)-й отрезок ранее уже не мог быть размещен.

На этом факте и основано доказательство теоремы. Вместо отрезка берем булев куб B^n и размещаем в нем шары $S_t(x) = \{y \in B^n : \rho(x,y) \le t\}$. Таким образом будет построено множество векторов V^*, которые соответствуют центрам размещенных шаров.

Построим V^*следующим образом:

Итерации:

$x^1 \to V^*$

$x^2 \to V^* \setminus S_t(x')$

…

$p)\ x^p \to V^* \setminus \bigcup_{i=1}^{p-1} S_t(x^i)$

В какой-то момент очередной шар уже не поместится. Расстояние между любыми точками таким образом построенной конструкции равняется $2t$, а шары радиуса $2t$ с центрами в этих точках покроют весь булев куб. Так как они могут пересекаться, то суммарное количество точек в этих шарах будет не меньше мощности куба, откуда и следует неравенство в теореме.

Теорема доказана.

Теорема (Плоткина). При условии, что 2d>n справедливо соотношение:

$$A(n,d) \le \frac{2d}{2d-n} \quad .$$

Доказательство. Рассмотрим множество булевых векторов $V = \{x^1, \ldots x^S\}$, попарные расстояния между которыми не менее d. Построим таблицу.

$G = \begin{vmatrix} x_1^1 & x_2^1 & \ldots & x_n^1 \\ x_1^2 & x_2^2 & \ldots & x_n^2 \\ . & . & . & . \\ x_1^S & x_2^S & \ldots & x_n^S \end{vmatrix}$, где k_1 , k_2 ,…,k_n – количество единиц в столбце.

Рассмотрим сумму расстояний между всеми векторами множества.

$$2\sum_{ij}\rho(x^i,x^j) = \sum_{ij}\sum_{t=1}^{n}|x_t^i - x_t^j| = \sum_{ij}\sum_{t=1}^{n}\left(x_t^i - x_t^j\right)^2 = \sum_{ij}\sum_{t=1}^{n}x_t^{i\,2} + \sum_{ij}\sum_{t=1}^{n}x_t^{j\,2} -$$

$$-2\sum_{ij}\sum_{t=1}^{n}x_t^i x_t^j = 2S\sum_{t=1}^{n}k_t - 2\sum_{t=1}^{n}k_t^{\,2} = 2\sum_{t=1}^{n}k_t(S-k_t)$$

Здесь каждое расстояние подсчитано 2 раза. Тогда $\sum_{i<j}\rho(x^i,x^j) = \sum_{t=1}^{n}k_t(S-k_t)$.

Заметим, что max $x(S\text{-}x)$ достигается при $x = S/2$. Воспользуемся этим. Тогда справедливо соотношение:

$$\sum_{t=1}^{n}k_t(S-k_t) \le \frac{nS^2}{4} \qquad (11.1)$$

Кроме того, по условию $\rho \ge d$. Поставим это неравенство в (11.1):

$\frac{nS^2}{4} \ge \sum_{i,j}\rho(x^i,x^j) \ge \frac{(S^2-S)d}{2}$. Отсюда следует nS≥2d(S-1) и

$S \le \frac{2d}{2d-n}$.

Теорема доказана.

11.2 Сводка результатов.

Рассмотренные в этих теоремах конструкции имеют разнообразное применение в комбинаторном анализе. Так как ниже будем их применять при передачи информации по каналу с шумом, то используем терминологию, принятую в теории кодирования.

Пусть B^n - булев куб размерности n.

Определение. Кодом *C* называется **произвольное** подмножество $C \subset B^n$.

Пусть M – мощность множества C. Элементы этого подмножества будем называть *кодовыми словами*.

Пусть k - минимальное целое, такое что $M \le 2^k$. (Ниже будет рассмотрен важный класс кодов, где всегда $M=2^k$).

Определение. Кодовой скоростью называется число $V(C) = \dfrac{\log M}{n}$.

Определение. Кодовым расстоянием называется число $d(c) = \min\limits_{\substack{x \neq y \\ x,y \in C}} \rho(x,y)$.

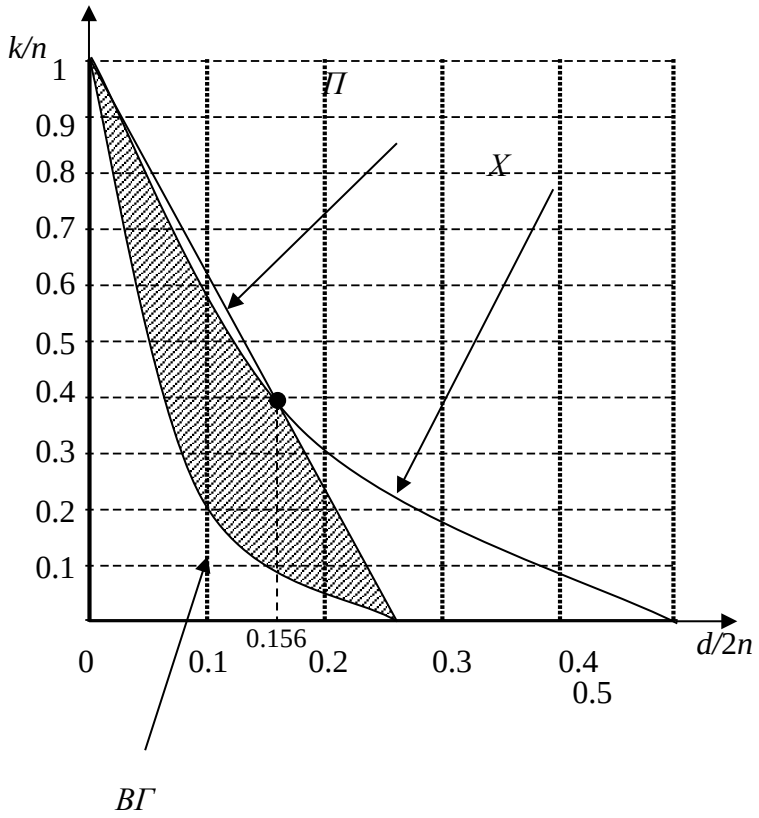
Так как расстояние Хемминга $\rho(x,y)$ $x, y \in B^n$ - это количество различных бит в *x* и *y*, а

||*x*|| - количество единиц в $x \in B^n$ (*норма* или *вес* *x*). Тогда очевидно, что

$$\rho(x,y) = \| x \oplus y \|.$$

Определение. (*n*,*M*,*d*)-код – это код *C*, такой, что |*C*| = *M*, $C = \{x^i\}$, $x^i \in B^n$, $d(C) = d$.

Сведем полученные в приведенных выше теоремах результаты на один график, по координатам которого даны отношения k/n и d/2n. (Этот график можно найти в любой книге по теории кодов, исправляющих ошибки, например, в [7]).



Соответствующие кривые, вытекающие из теорем, обозначены: ВГ- Теорема Варшамова-Гильберта, П- Теорема Плоткина, Х - Теорема Хэмминга.

Комментарий к графику.

Из графика видно, что при постоянном отношении k/n и росте размерности куба границы для отношения d/n стремятся константе, не зависящей от этой размерности.

(n,M,d)-коды существуют только для тех значений параметров, которые лежат в заштрихованной области. Однако, не для любой точки в этой области (а каждой такой точке соответствует фиксированная тройка параметров n,M,d) известен (построен) (n,M,d)-код.

Ниже будут рассмотрены несколько примеров известных кодов, в частности коды Хэмминга и коды Боуза-Чоудхури-Хоквингема (БЧХ-коды). Они при больших скоростях передачи лежат на границе Хэмминга и являются оптимальными. Для малых скоростей БЧХ-коды лежат на границе Плоткина.

При средних скоростях для БЧХ-кодов отношение d/n стремится к 0, когда n неограниченно растет.

Неизвестны коды, для которых d/n остается конечным при росте n и фиксированном k/n, хотя граница Варшамова-Гильберта дает основание считать, что такие коды должны быть.

В следующих лекциях будет доказана вторая теорема Шеннона (теорема Шеннона для канала с шумом). Из нее следует, что при любой скорости k/n, меньшей пропускной способности канала, существуют коды, вероятность ошибки декодирования в которых может быть сделана сколь угодно близкой к нулю.

В целом, существующие коды, с теоретической точки зрения, далеки от оптимальных, поэтому теория кодирования до сих пор дает возможность исследователям получать интересные результаты.

11.3 Упражнения.

Пусть A(n,d) – максимальное число точек в коде из B^n с минимальным расстоянием d.
Пусть B(n,d) – максимальное число точек в коде из B^n такое, что расстояние между любой парой не больше d.

1. Доказать неравенство $A(n,2t+1)\cdot B(n,2t)\le 2^n$.

2. Проверить существование (8,12,3)-кода. Проверить существование (12,200,3)-кода.

3. При каких параметрах граница Джоши лучше границы Хэмминга?

4. Доказать соотношения:

a) $\sum_{x,y\in E^n}\rho(x,y)=n2^{2n-2}$;

b) $\sum_{x,y\in E_k^n}\rho(x,y)=n\binom{n-1}{k-1}\binom{n-1}{k}$;

c) $\sum_{x,y\in S_t^n(\alpha)}\rho(x,y)=nS_{n-1}^{t-1}\left(S_{n-1}^{t-1}+n+1\right)$, где $S_n^r=\sum_{i=0}^{r}\binom{n}{i}$.

12. Лекция 12. Канал с ошибкой замены: двоичный симметричный.

12.1 Введение.

Выше был рассмотрен случай, когда при передаче информации происходит потеря букв передаваемых слов.

А что будет, если символы искажаются, причем подобная ошибка может быть многократной. В предыдущем курсе алгебры у вас был раздел *алгебраической теории кодирования*, где основное внимание уделялось алгебраическим методам конструирования способов защиты от таких ошибок.

Мы же здесь рассмотрим два вопроса.

1. При каких соотношениях между параметрами канала и используемого для защиты от шума кода с вероятностью близкой к единице возможна передача информации без потерь. (Вторая теорема Шеннона).
2. Примеры важных комбинаторных характеристик кодовых конструкций.

В предыдущей лекции мы уже говорили, что рассмотренные там конструкции на метрических множествах легко интерпретируются и широко используются в теории кодирования. Но это совсем не единственное поле их применения, поэтому исследование комбинаторных характеристик таких конструкций – это важный и практически значимый вопрос дискретного анализа. Но объем курса позволяет, в лучшем случае, привести только примеры таких характеристик.

Для построения и исследования кодов необходимо иметь информацию о структуре кода (расположении его точек в булевом кубе), а для этого необходимо ввести метрику.

Вся эта лекция будет посвящена подготовке к доказательству второй теоремы Шеннона. Следующая лекция – это доказательство теоремы, а оставшиеся лекции – примерам комбинаторных характеристик кодов.

12.2Принципы построения кодов, исправляющих ошибки.

Так как ваш учебный план, возможно, еще не содержал теории вероятностей, то везде ниже будет, по сути, использоваться та же конструкция, что и при доказательстве теоремы о длине минимального теста для «почти всех» матриц.

Дано множество W. Задано свойство S, которым обладает часть элементов этого множества W_s. Под вероятностью того, что объект обладает свойством s будем понимать отношение $|W_s|/|W|$ или его асимптотику.

Итак, у нас есть двоичный симметричный канал. (См. определение выше). Он характеризуется единственным параметром p – вероятностью ошибки при передаче символа.

Определение. Пропускная способность канала – это число $C(p) = 1 - H(p)$, где

$H(p) = p\log\frac{1}{p} + (1-p)\log\frac{1}{1-p}$ - энтропия канала.

Как раз для случая p=1/2 энтропия равна единице, что соответствует полной неопределенности (абсолютному беспорядку), а пропускная способность канала нулевая.

Если p>1/2, то поменяем 0 и 1 местами в определении конала. Поэтому всюду в дальнейшем считаем, что p<1/2.

В качестве защиты от ошибок мы используем некоторую конструкцию. Это (n,M,d) – код. Как мы видим, у него три параметра. Пусть t - максимальный радиус шаров с центрами в кодовых точках таких, что они не пересекаются. Очевидно, что t выражается через d и наоборот.

Схемы кодирования могут быть самые разные. Рассмотрим одну из самых известных. (Она считается наиболее естественной и соответствующей практическим задачам.)

В ее основе лежит два принципа:

1. **Принцип наибольшего правдоподобия**. Этот принцип вытекает из очевидного соотношения

$$p^n < p^{n-1}(1-p) < p^{n-2}(1-p)^2 < \ldots < (1-p)^n$$

и говорит о том, что вероятность k ошибок в кодовом слове меньше, чем вероятность s ошибок для всех s>k.

2. **Принцип избыточности**.

Мы в нашем рассмотрении не затрагиваем протокольные вопросы, а ограничиваемся чисто алгоритмическими, т.е кодер и декодер в процессе передачи не ведут между собой никакого диалога (например, не могут перезапрашивать один раз уже переданные символы и т.п.) В этом случае алгоритм декодирования по принятой информации должен исправлять ошибки, которые возникли при передаче. Но, если кодер просто передает в канал информацию, подлежащую передаче, ничего к ней не добавляя, то у декодера нет никакой возможности исправить ошибки. Таким образом, принцип избыточности заключается в том, что для защиты передаваемой информации используется дополнительная информация. И тут возникает уже знакомая конструкция: нужно уметь отделять кодовые точки друг от руга определенным расстоянием. А это и будут (n,M,d)-коды.

Мы считаем, что все кодовые слова имеют одинаковую длину. (Это вектора пространства B^n). Информация, подлежащая передаче может быть представлена в виде последовательности двоичных слов длины k (k<n). Кодер берет эти слова и преобразует в слова длины n. Здесь и возникает избыточность за счет (n-k) символов.

Если кодер просто добавляет к словам, подлежащим передаче эти (n-k) символов, то они называются проверочными символами, а исходные k символов – информационными.

Так построенный код называется *систематическим кодом*.

Однако, надо заметить, что далеко не всегда *избыточность* реализуется столь простым способом. Поэтому принцип избыточности заключается в том, что для защиты от ошибок в канале к информации, подлежащей передаче, что-то добавляется. За счет этой добавки декодер имеет возможность выяснить, произошли ли ошибки и, если произошли, то исправить их.

Проиллюстрируем сказанное.

Пусть, например, информация, подлежащая передаче, - это битовый поток, подающийся на вход кодера. Кодер нарезает ее в виде последовательности слов длины k. Таким

образом можно считать, что передаче в канал подлежат все возможные слова длины k, т.е. вектора пространства B^k, таких словам $M=2^k$. (Случай $2^{k-1}<M<2^k$ –можно “технически свести» к рассматриваемому, поэтому наше предположение не ограничивает общности). Кодер преобразует вектора пространства B^k в вектора пространства B^n. Таким образом реализуется принцип избыточности.

В общем случае мы имеем M слов, подлежащих передаче, которые могут быть идентифицированы с помощью битовых векторов, длины меньшей n.

Обозначим эти слова через $\alpha^1,\alpha^2,\ldots,\alpha^M$, а сопоставленные им вектора пространства B^n (кодовые вектора) через $x^1,x^2,\ldots,x^M$.

Определение. Отображение φ , ставящее в соответствие α^i, i=1,…,M, вектор из B^n и называется *кодированием* или просто *кодом*.

Из контекста всегда видно, что понимается под *кодом*: *множество* или *отображение*.

Так как мы рассматриваем ситуацию защиты информации от помех в канале, то для таких кодов используются следующие названия: корректирующий код, код с исправлением ошибок, код с обнаружением ошибок и т.п.

К нему предъявляются естественные требования *дешифруемости* и *неизбыточности*. То есть, если $\varphi(\alpha^i)=x^i$, то кодер и декодер всегда «знают» (умеют вычислять), что $\varphi^{-1}(x^i)=\alpha^i$. Но не это является основной проблемой при построении кодов, исправляющих ошибки. Дело в том, что переданное кодером в канал слово x^i в результате искажения может превратиться в какое-то слово Y^i (это тоже вектор из B^n).

Основная задача кодирования. Отображение φ должно быть построено таким образом, чтобы можно было сконструировать алгоритм преобразования Y^i в α^i. И это должно приводить к исправлению ошибок. Поэтому под *декодированием* (*дешифруемостью*) в коде, исправляющем ошибки понимается именно данное преобразование. (Обозначим это преобразование через ψ).

Построение каждого такого кода – это отдельное достижение в математике, поэтому принято кодам давать имена их создателей: код Хэмминга, код Боуза- Чаудхури-Хоквингема (БЧХ-код), коды Гоппа и т.д.

Интуитивно ясно, что чем хуже канал (больше вероятность ошибки), тем большая требуется избыточность. Поэтому каждый код может исправить только ограниченное количество ошибок в слове определенной длины. Например, если код может исправлять не более двух ошибок в слове длины n=10 , то при передаче информации по каналу с p=1/3 этот код не сможет всегда исправлять все ошибки (ниже будет показано, что среднее количество ошибок в слове в таком канале равно 3). Если же этот код используется в канале с p=1/1000, то такой код свои задачи выполнит (с «вероятностью, близкой к единице»). Таким образом, коды подбираются в соответствии с характеристиками канала.

Пример.

Мажоритарное кодирование – схема, где передатчик просто повторяет слово, подлежащее передаче в канал, $p=2q+1>1$ раз. Приемник – декодер берет экземпляров полученного

слова и каждую позицию в слове дешифрует в тот символ (0 или 1). Который встречается в большей части экземпляров.

Например, двухбитовой слово $a = 11 \rightarrow 11|10|11|00|11|11|11$ – передали по каналу 7 раз.

На первой позиции 6 раз «1» и 1 раз «0» $\Rightarrow$ $a = 1x$.

На второй позиции 5 раз «1» и 2 раза «0» $\Rightarrow$ $a = 11$.

Избыточность этой схемы очень большая.

12.3Декодирование на основе таблицы декодирования.

Декодирование (реализация вышеупомянутого преобразования ψ) может производиться различными способами. Простейшим и самым наглядным является его реализация с помощью таблицы декодирования. Рассмотрим две ситуации.

1. Пусть нам удалось так построить алгоритм кодирования, что каждую кодовую точку можно было окружить шаром радиуса t так, чтобы эти шары не пересекались. Тогда *Таблица декодирования* строится следующим образом. Первая строка таблицы – все кодовые вектора. В столбце под кодовым вектором лежат все вектора B^n, расположенные в шаре радиуса t с центром в этом кодовом векторе. Все оставшиеся вектора B^n размещаются по столбцам таблицы согласно какому-то эвристическому правилу. (Например: a) все в первый столбец; b) все случайным образом; c) в столбец с ближайшим кодовым словом, а в случае равноудаленности от нескольких кодовых векторов – к тому, номер столбца которого наименьший. И т.п.) Декодирование осуществляется следующим образом: полученный вектор преобразуется в первый вектор столбца, в котором он расположен.
2. Пусть нам не удалось так построить алгоритм кодирования, что каждую кодовую точку можно было окружить шаром радиуса t так, чтобы эти шары не пересекались. Но *Таблица декодирования* строится так же, как и в предыдущем случае. Декодирование осуществляется тоже так же: полученный вектор преобразуется в первый вектор столбца, в котором он расположен.

Совершенно очевидно, что в первом случае есть какая-то логика и надежда на декодирование. А во втором случае - все хуже. В том смысле, что ошибки декодирования, по-видимому, будут чаще.

Мы привели этот случай для того, чтобы проиллюстрировать доказательство приведенной ниже теоремы Шеннона, в котором кодом будем любая случайно выбранная совокупность векторов, поэтому и для нее должны бать фиксированы правила декодирования.

Пример:

C: $a = 11 \rightarrow x^1 = \underline{11}000$

$b = 00 \rightarrow x^2 = \underline{00}110$

$c = 10 \rightarrow x^3 = \underline{10}011$

$d = 01 \rightarrow x^4 = \underline{01}101$

Таблица декодирования.

11000	**00110**	**10011**	**01101**
11001•	00111•	10010•	01100•
11010•	00100•	10001•	01111•
11100•	00010•	10111•	01001•
10000•	01110•	11011•	00101•
01000•	10110•	00011•	11101•
11110	00000	01011	10101
01010	10100	11111	00001

Источник Кодер B^5

a x^1

b x^2 $\rightarrow$ канал $Y \in B^5$

c x^3 C – (5,4,3)-код

d x^4

Всего в таблице декодирования $|B^5| = 32$ вектора. В каждом столбце находятся все пять векторов, которые расположены на расстоянии ρ=1 от кодового вектора (это первый вектор данного столбца). Кроме того, в каждом столбце находится по два вектора, которые расположены на расстоянии ρ=2 от двух кодовых векторов.

Получается, что в нашей схеме декодирования кодовые вектора окружены шарами единичного радиуса.

Таким образом, если получено любое слово из первых пяти строк таблицы, то декодирование произойдет правильно. Если же взять другое слово, то декодирование произойдет либо правильно, либо нет.

12.4Схема кодирования по принципу наибольшего правдоподобия на основе принципа избыточности.

Рассмотрим теперь приведенный выше подход к построению кода более подробно.

Пусть мы имеем ситуацию 1. Шар B^n, в нем 2^n точек. $M < 2^n$. Стараемся, чтобы каждую из этих точек можно было окружить шаром радиуса t так, чтобы эти шары не пересекались. Заметим, что из того, что $p < ½$, следует:

$$p^n < p^{n-1}(1-p) < p^{n-2}(1-p)^2 < \ldots < (1-p)^n.$$

Это значит, что вероятность того, что ошибок не будет, больше, чем того, что будет одна ошибка и т.д. Это единственное обоснование схемы кодирования по наибольшему правдоподобию.

Дешифруем Y^i. В шаре радиуса d с центром в точке x^i на расстоянии 1 стоят все векторы, которые получились бы из x^i в случае одной ошибки, на расстоянии 2 все векторы с двумя ошибками и т.д.

Возможны 3 случая:

1. Пусть шары с центрами в кодовых точках имеют радиус d и не пересекаются. Y^i оказалась в шаре с центром в x^i – произошло не более, чем d ошибок, $\rho(x^i, Y^i) \le d$;
2. Y^i не попадает ни в один из шаров, $\rho(x^j, Y^i) > d$ для любого j; (в этом случае алгоритм декодирования должен предусматривать правило сопоставления Y^i какому-то из x^j).
3. $\rho(x^j, Y^i) \le d$ (попала не в этот шар, а в соседний).

Если имеет место первый случай, то $Y^i \to x^i$. Если третий, то $Y^i \to x^j$, декодирование неверное.

Во втором случае обычно декодируют Y^i в x^j, тоже может быть ошибка декодирования.

Пусть мы имеем ситуацию 2. Тогда у нас первого случая не будет вообще, а будут только два остальных. Но если ближайшая к полученному кодовая точка единственна, то вероятность правильности декодирования в эту точку больше, чем в любую другую.

Канал мы будем считать источником Бернулли с точки зрения вероятности ошибки в переданном символе. Действительно, эта вероятность не зависит от места символа в битовой последовательности и от того, были ли ошибки в предыдущих и последующих символах. Если канал не подчиняется этим требованиям (то это уже будет не двоичный симметричный канал, а некоторый канал со специальными свойствами), то все нижеследующие рассуждения не справедливы.

Пусть $P(x)$ – вероятность ошибки декодирования при передаче в двоичный симметричный канал кодового слова x. (А $\Pi(x)$ – вероятность правильного декодирования.) (Ниже при доказательстве утверждений будут подробно обсуждаться формулы для этих величин.)

Обозначим через $P_c = 1\backslash M \sum P(x)$ – среднюю ошибку на одно кодовое слово в коде C (суммирование берется по всем кодовым словам). Таким образом, число $P_c = 1\backslash M \sum P(x)$ является характеристикой кода C.

Число различных кодов мощности M в B^n: равно C_{2n}^M .

Пусть L – множество всех таких кодов.

Пусть $\Pi(x)$ – вероятность правильного декодирования переданного слова x. Тогда для кода C имеем $\Pi_c = 1\backslash M \sum \Pi(x)$.

Тогда $P_c + \Pi_c = 1$.

Обозначим через $P^*(M, n, p) = \min P_c$, где минимум берется по всем кодам из множества L. То есть $P^*(M, n, p)$ – характеристика некоторого **существующего** кода, являющаяся наилучшей из возможных с точки зрения ошибки декодирования.

Очевидно, что

$$P^*(M, n, p) \le \frac{1}{C_{2^n}^M} \sum_{x^i \in C} \mathrm{P}_c(x^i).$$

Обозначим через $\Pi^*(M,n,p)= \max \Pi_c$, где максимум берется по всем кодам из множества L. То есть $\Pi^*(M,n,p)$ – характеристика некоторого **существующего** кода, являющаяся наилучшей из возможных с точки зрения ошибки декодирования.

Очевидно, что

$$\Pi^*(M, n, p) \ge \frac{1}{C_{2^n}^M} \sum_{x^i \in L} \Pi_c(x^i).$$

12.5Корректирующие способности кодов.

Очевидно, что параметры (n,M,d)-кода зависят друг от друга, отсюда следует, что для каких-то значений этой тройки параметров коды существуют, а для каких-то нет. В приведенном выше примере декодирования на основе таблицы декодирования был построен (5,4,3)- код. Уже из определения таблицы декодирования и описанной выше схемы следует, что, например, (5,4,5)-код построить нельзя, так как для любого (5,4, d)-кода в столбце таблицы декодирования будет 7 элементов, а в шаре радиуса 2, который требуется для (5,4,5)-кода, 15 точек.

В этом разделе будут приведены несколько утверждений, которые помогают ответить на вопрос о существовании кода с заданными параметрами.

Но для начала сформулируем очевидное утверждение.

Определение. Код называется кодом, исправляющим t ошибок, если алгоритм декодирования правильно декодирует все слова, в которых при передаче по каналу произошло не более t ошибок.

Утверждение. (n,M,d)-код – код с исправлением $\left\lfloor \frac{d-1}{2} \right\rfloor$ ошибок.

Замечание. Есть еще коды, обнаруживающие ошибки.

Опр. Код называется кодом, обнаруживающим t ошибок, если в случае, когда

произошло не более t ошибок, декодер выдает сообщение об ошибке.

Утверждение. (n,M,d)-код обнаруживает $(d-1)$ ошибку.

13. Лекция 13. Теорема Шеннона для канала с шумом.

13.1Введение.

Есть несколько теорем Шеннона. Одна была рассмотрена выше (теорема Шеннона для канала без шума). Здесь мы докажем еще одну теорему Шеннона и об одной просто упомянем. (Эти теоремы известны как теоремы Шеннона для канала с шумом.)

В теореме Шеннона для канала с шумом будет показано, что существуют коды, когда для любого $\varepsilon>0$ при $V(C)< 1- H(p)$ вероятность ошибки декодирования не более ε.

Рассмотрим некоторый код C = $\{x^1 \ldots x^M\}$, где $\{x^1 \ldots x^M\}$– кодовые слова (вектора B^n).

Применяем декодирование в ближайшее слово (по принципу наибольшего правдоподобия и согласно приведенной выше схеме декодирования, а ,в случае нескольких вариантов, в кодовое слово с наименьшим номером). Обозначим через T(C) таблицу декодирования.

Передача

a	x^1
b	x^2
…	…
…	x^M

Прием

T(C) – таблица

x^1	…	…	x^M
..	..	..	..
..	..	..	..

$x^i \in B^n$

Зададим некоторое ρ>0. Тогда в ситуации 1 таблица $T_\rho(C)$ состаляется так, чтобы под каждым кодовым словом $x^i \in B^n$ в столбце находились все слова из $S_\rho(x^i)$, а в ситуации 2 работают эвристические правила, примеры которых приведен выше.

Замечание.

Подчеркнем еще раз, что кодом называется **любое** подмножество булева куба B^n мощностью M. Поэтому не любой код будет кодом, исправляющим ошибки. Но к любому коду можно формально применить описанную выше схему декодирования. Все точки кода окружаем шарами радиуса ρ, а полученный вектор декодируем в центр шара. Если же вектор попадает в несколько кодовых шаров, то, возможно, что произошла ошибка декодирования. Правило, по которому в этом случае декодер производит декодирование несущественно. Например, применяется какое-нибудь дополнительное правило декодирования: в случайный, в ближайший и т.п.

Возьмем точку x в B^n и шар $S_{[pn]}(x)$ радиуса $[pn]$с центром в этой точке. Оценим число точек в нем.

Лемма 13.1. Справедливо неравенство

$$|S_{[pn]}(x)| \le 2^{nH(p)}$$

Доказательство. $|S_{[pn]}(x)| = \sum_{i=0}^{[pn]} C_n^i$.

Вспомним, что $0 \le p < 1\backslash 2$. Согласно принципу наибольшего правдоподобия – декодирование в ближайшее кодовое слово – выполняется соотношение

$(1-p)^n \ge p(1-p)^{n-1} \ge \ldots \ge p^k(1-p)^{n-k}$.

Отсюда следует.

$$1 = \left(p + (1-p)\right)^n = \sum_{i=0}^{n} C_n^i p^i (1-p)^{n-i} \left(\ge\right)$$

$$\left(\ge\right) \sum_{i=0}^{[pn]} C_n^i p^i (1-p)^{n-i} \ge p^{[pn]}(1-p)^{n-[pn]} \sum_{i=0}^{[pn]} C_n^i,$$

где $\sum_{i=0}^{[pn]} C_n^i$ - мощность шара радиуса [pn] .

Тогда $|S_{[pn]}(x)| \le (p^{[pn]}(1-p)^{n-[pn]})^{-1} = 2^{-\log p^{[pn]}(1-p)^{n-[pn]}} \le 2^{nH(p)}$.

Лемма доказана.

13.2Комбинаторное доказательство.

Теорема 13.1. (Вторая теорема Шеннона).

Для любого R: $0 < R < C(\rho)$, такого что $M=2^{[nR]}$, выполняется соотношение:

$$П^*(M,n,p) \underset{n\to\infty}{\to} 1.$$

Доказательство.

Пусть имеется код $V_k \in L=\{V_1, V_2, ..., V_{C_{2^n}^M}\}$. Напомним, что $П(V_k)$ – вероятность правильного декодирования для V_k .

Рассмотрим 3 функции:

$$\xi_k(a,b) = \begin{cases} 1, \text{ если } b \text{ дек. в } a \\ 0, \text{ в против. случае} \end{cases}$$

$$\eta_k(a,b) = \begin{cases} 1, \text{ если } a \in V_k \\ 0, \text{ если } a \notin V_k \end{cases}$$

$$\delta^i(a,b) = \begin{cases} 1, \text{ если } \rho(a,b) = i \\ 0, \text{ в против. случае} \end{cases}$$

Пусть $A_k(i)$ – число векторов в B^n, находящихся на расстоянии i от некоторого кодового слова кода V_k и декодируемых в это кодовое слово.

Тогда $П(V_k) = \frac{1}{M}\sum_{i=1}^{n} A_k(i) p^i (1-p)^{n-i}$ - вероятность правильного декодирования V_k. Здесь $A_k(i) = \sum_{a,b\in B^n} \xi_k(a,b)\eta_k(a,b)\delta^i(a,b)$. Действительно, каждое слагаемое в этой сумме равно единице, если выполняются все три условия:

1) $a \in V_k$,
2) расстояние от a до b равно i,
3) b декодируется в a.

В формуле для вероятности правильного декодирования $П(V_k)$ параметр $A_k(i)$ – это число точек из B^n, которые находятся на расстоянии i от некоторого кодового слова и декодируются в это кодовое слово.

Всего таких кодов $C_{2^n}^M$: $V_1, V_2, ..., V_{C_{2^n}^M}$, $|V_i| = M$.

Усредняем вероятность правильного декодирования по всем кодам и получаем среднюю вероятность правильного декодирования

$$\overline{P}(M,n,p)=\frac{1}{C_{2^n}^{M}M}\sum_{k=0}^{C_{2^n}^{M}}\sum_{i=0}^{n}\sum_{a,b\in B^n}\xi_k(a,b)\eta_k(a,b)\delta^i(a,b)\,p^i(1-p)^{n-i}=$$

$$=\frac{1}{C_{2^n}^{M}M}\sum_{i=0}^{n}p^i(1-p)^{n-i}\sum_{a,b\in B^n}\delta^i(a,b)\sum_{k=1}^{C_{2^n}^{M}}\xi_k(a,b)\eta_k(a,b) \qquad (13.1)$$

Рассмотрим саму внутреннюю сумму в последней формуле приведенных выше выкладок. Эта сумма равна числу кодов V_k таких, что $a\in V_k$, и b декодируется в a. Заметим, что точка b обязательно декодируется в a тогда и только тогда, когда в коде V_k нет кодовых точек расположенных к b ближе, чем точка a. Это означает, что в шаре (обозначим его S(a,b)) радиуса ρ(a,b) с центром в точке b нет других точек кода V_k. Поэтому, если точки a,b фиксированы, то остальные (M-1) точку кода мы можем выбирать произвольным образом из множества $\Gamma= B^n\setminus S(a,b)$. Мощность этого множества $2^n-|S_\rho(x)|$, где $|S_\rho(x)|$количество точек из B^n в шаре радиуса ρ. (Так как это количество не зависит от x, а зависит только от n и ρ, то, чтобы подчеркнуть этот факт, ниже при необходимости мы наряду с обозначением $|S_\rho(x)|$ будем пользоваться и другим -$|S_\rho^n|$). Отсюда следует, что упомянутая внутренняя сумма равна $C^{M-1}_{2^n-\sum_{i=0}^{\rho}C_n^i}$.

Отметим два комбинаторных факта и продолжим начатую выше цепочку соотношений.

1) Число пар точек из B^n, лежащих на расстоянии i равно $2^nC_n^i$.
2) Справедливо известное комбинаторное равенство

$$\frac{1}{C_{2^n}^{s}}=\frac{s}{2^nC_{2^n-1}^{s-1}}.$$

Далее из (13.1) имеем

$$(=)\ \frac{2^n}{sC_{2^n}^{s}}\sum_{i=0}^{n}C_n^ip^i(1-p)^{n-i}C_{2^n-|S_i(0)|}^{s-1}=\frac{1}{C_{2^n-1}^{s-1}}\sum_{i=0}^{n}C_n^ip^i(1-p)^{n-i}C_{2^n-|S_i(0)|}^{s-1}. \qquad (13.2)$$

Мы доказали одно из основных равенств, которое будем использовать ниже.

Отметим несколько моментов:

1) Функция $F(x)=C_n^x$ выпукла вверх.
2) Тогда функция $F^*(x)=C_{n-x}^k$ выпукла вниз.
3) Сумма выпуклых функций выпукла.

4) Выпуклая функция от выпуклой функции выпукла.

5) Отсюда следует, что функция f(i)= $\begin{pmatrix} 2^n - \sum_{k=0}^{i} C_n^k \\ M-1 \end{pmatrix} = \begin{pmatrix} \sum_{k=i+1}^{n} C_n^k \\ M-1 \end{pmatrix}$ является функцией, выпуклой вниз.

6) Вспомним **неравенство Йенсена**. Пусть даны числа $\alpha_1 \ldots \alpha_n$, $\alpha_i > 0$, $\sum \alpha_i = 1$. Тогда для выпуклой вниз функции f(x) справедливо неравенство.

$$\sum_{i=1}^{n} \alpha_i f(x_i) \geq f(\sum_{i=1}^{n} \alpha_i x_i).$$

7) В качестве α_i возьмем $C_n^i p^i (1-p)^{n-i}$. Очевидно, что $\sum_{i=0}^{n} C_n^i p^i (1-p)^{n-i} = 1$, а $0 \leq C_n^i p^i (1-p)^{n-i} \leq 1.$

Используя 1-7, получим

$$\sum_{i=0}^{n} C_n^i p^i (1-p)^{n-i} f(i) \geq f\left(\sum_{i=0}^{n} i C_n^i p^i (1-p)^{n-i} \right). \quad (13.3)$$

Далее используем очевидные соотношения

$$\sum_{i=0}^{n} i C_n^i p^i (1-p)^{n-i} = np, \quad \frac{1}{C_{2^n}^M} = \frac{M}{2^n} \frac{1}{\begin{pmatrix} 2^n - 1 \\ M-1 \end{pmatrix}}. \quad (13.4)$$

Заметим далее, что максимальная вероятность больше средней, т.е.

$$П^*(M,n,p) \geq \overline{P}(M,n,p).$$

Из этого факта и соотношений (13.2), (13.3), (13.4) получаем:

$$П^*(M,n,p) \geq = \frac{1}{\begin{pmatrix} 2^n - 1 \\ M-1 \end{pmatrix}} \sum \begin{pmatrix} n \\ k \end{pmatrix} p^i (1-p)^{n-i} f(i) \geq \frac{1}{\begin{pmatrix} 2^n - 1 \\ M-1 \end{pmatrix}} f(np).$$

Далее

$$(C_{2^n-1}^{M-1})^{-1} f(np) = \frac{C_{2^n - S_{np}^n}^{M-1}}{C_{2^n-1}^{M-1}} = \frac{(2^n - |S_{np}^n|)(2^n - |S_{np}^n| - 1)\ldots(2^n - |S_{np}^n| - M)}{2^n(2^n-1)\ldots(2^n - M)} = \prod_{k=0}^{M} \left(1 - \frac{|S_{np}^n|}{2^n - k} \right) >$$

$$> 1 - |S_{[np]}^n| \sum_{k=0}^{M} \frac{1}{2^n - k} = 1 - 2^{-n} |S_{[np]}^n| \sum_{i=0}^{M} \sum_{k=1}^{\infty} (1 + \frac{i^k}{2^{kn}}) \geq 1 - 2^{-n} |S_{[np]}^n| \sum_{i=0}^{M} \sum_{k=1}^{\infty} (1 + \frac{i^k}{2^{kn}})$$

Теперь устремляем $n \to \infty$ и раскладываем в ряд, оставляя главные члены каждого слагаемого. Получаем

$$1-2^{-n}\,|\,S^n_{[np]}\,|\sum_{i=0}^{M}\sum_{k=1}^{\infty}(1+\frac{i^k}{2^{kn}}) = 1\text{-}\,(\,2^{-n}\,|\,S^n_{[np]}\,|\,M + 2^{-2n}\,|\,S^n_{[np]}\,|\,M(M+1)/2 + 2^{-3n}\,|\,S^n_{[np]}\,|\sum_{i=0}^{M} i^2 + \ldots) =$$

$$=1\text{-}\,2^{-n}\,|\,S^n_{[np]}\,|\,M + \mathrm{O}(\,2^{-2n}\,|\,S^n_{[np]}\,|\,M^{\,2})\,,$$

Что асимптотически при n→∞ равно

$$1-|\,S^n_{[np]}\,|\,\frac{M}{2^n}.$$

Вспомним лемму (13.1):

$$|S_{[pn]}(x)| \le 2^{nH(p)}.$$

Кроме того, заметим, что по условию: $0 < R < C(\rho)$, $M=2^{[nR]}$.

Отсюда получаем

$$1-|\,S^n_{[np]}\,|\,\frac{M}{2^n} \ge 1-\frac{M}{2^n}\,2^{nH(p)} \ge 1-\frac{2^{n[R+H(\,p)]}}{2^n} = 1\text{-}\,2^{\,\mathrm{n(R\text{-}C(p))}}.$$

Таким образом мы доказали, что при $0 < R < C(\mathrm{p})$ и n→∞ справедливо неравенство

$$\Pi^*(M,n,p) > 1\text{-}\,2^{\,\mathrm{n(R\text{-}C(p))}}.$$

Но это значит, что $\Pi^*(M,n,p) \to 1$ при n→∞ .

Теорема доказана.

13.3 Дополнения и замечания.

Теорема Шеннона – это теорема существования, и она не дает методов построения кодов. Разбор, классификация и общее представление совокупности таких методов – предмет отдельной дисциплины – *теории кодирования*.

А что будет, если R>C(p) ? На этот вопрос отвечает третья Теорема Шеннона.

Теорема 13.2. (Третья теорема Шеннона.) Для любого R: $R > C(\mathrm{p})>0$, такого что $M=2^{[nR]}$, не существует кода C, для которого бы выполнялось соотношение:

$$\Pi_X\,(M,n,p) \to 1 \quad \text{при } n\to\infty.$$

Мы оставим эту теорему без доказательства.

Доказательство этой теоремы базируется на нескольких фактах (см. [7]). В лемме 13.1 было доказано, что $|B_{pn}(x)| \le 2^{nH(p)}$. На самом деле, справедлива более точная оценка: $|B_{pn}(x)| \sim 2^{nH(p)}$ (при $n \to \infty$). И это асимптотическое равенство достигается для *совершенных кодов* (см. ниже). В итоге получим, что

$$k = n - n\,H(p)$$

для совершенного кода. А отсюда можно получить, что максимум вероятности правильного декодирования достигается именно на совершенном коде. После этого в качестве кода C при доказательстве теоремы можно брать только совершенный код. А это уже конструкция с определенными свойствами, которые и используются для доказательства теоремы.

13.4Вероятностное доказательство второй теоремы Шеннона.

Это необязательное дополнение к лекциям для тех, кто уже знаком с самыми азами теории вероятностей. Базируется на лекциях [8].

13.4.1 Факты из теории вероятности.

Пусть ξ - случайная величина. $E(\xi)$ - мат. ожидание. $D(\xi)$ - дисперсия.

Теорема 13.3. (Неравенство Чебышева) Справедливо неравенство:

$$P\left(|\xi - E(\xi)| > \varepsilon\right) \le \frac{D(\xi)}{\varepsilon^2}.$$

Ошибки в канале моделируются испытаниями Бернулли, вероятность успеха *p* (неудача 1-*p*). Есть *n* испытаний. (Удачное испытание – произошла ошибка, неудачное – не произошла. Так как передается n символов, то мы получаем n испытаний во время передачи кодового слова.)

В случае источника Бернулли справедливы равенства: $E(\xi) = np$ и $D(\xi)=np(1-p)$.

Первое из них очевидно, а ниже приведено краткое доказательство второго.

$$D(\xi) = \sum_{m=0}^{n} (m-np)^2 P(m) = \sum_{m=0}^{n} m^2 P(m) - 2np\sum_{m=0}^{n} mP(m) + n^2 p^2 \sum_{m=0}^{n} P(m) =$$

$$= \sum_{m=2}^{n} m(m-1)P(m) + np - n^2 p^2 = \sum_{m=2}^{n} \frac{n!}{(m-2)!(n-m)!} p^m (1-p)^{n-m} + np - n^2 p^2 =$$

$$= p^2 n(n-1)\sum_{m=0}^{n-2} \frac{(n-2)!}{m!(n-m-2)!} p^n (1-p)^{n-m-2} + np - n^2 p^2 = p^2 n(n-1) + np - n^2 p^2 =$$

$$= np(1-p).$$

Пусть $\xi = (e_1 \ldots e_n)$ - вектор ошибок. Он подчиняется распределению Бернулли, поэтому матожидание и дисперсия такого вектора $E(\xi) = np$ и $D(\xi)=np(1-p)$.

Рассмотрим некоторый код C = $\{x^1 \ldots x^M\}$, где $\{x^1 \ldots x^M\}$– кодовые слова (вектора B^n).

Применяем декодирование в ближайшее слово (по принципу наибольшего правдоподобия и согласно приведенной выше схеме декодирования, а ,в случае нескольких вариантов, в кодовое слово с наименьшим номером). Обозначим через T(C) таблицу декодирования.

Пусть в канал передается вектор x^i, а в декодер поступил вектор Y.

$$x^i \rightarrow \underline{\text{канал}} \rightarrow \text{Y}$$

Считаем, что $Y = x_i + e$, где e – вектор ошибок.

Пусть τ – случайная величина, равная $|e|$ (количество ошибок).

Т.к. так как относительно этой величины мы имеем распределение Бернулли, то $E(\tau)$ = np, $D(\tau) = np\,(1-p)$.

Вводится некоторый параметр $b = \sqrt{\frac{np(1-p)}{\varepsilon/2}}$, который носит чисто технический характер.

Грубо говоря, если на одно слово приходится np ошибок, то мы будем окружать кодовые слова шарами радиуса ρ =[np+b], где b>0 – некоторая «маленькая» добавка

Лемма 13.2. Справедливо соотношение

$$P\{\tau > \rho\} \le \varepsilon\backslash 2$$

для любого заранее заданного ε.

Доказательство. Применим неравенство Чебышева:

$$P\{|\tau - E(\tau)| > \varepsilon\} \le D(\tau)/\varepsilon^2.$$

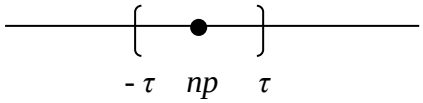
Положим $\rho = [E(\tau) + b]$, тогда

$$P\{|\tau - np| > b\} \le D(\tau) / b^2.$$

Отсюда следует

$$P\{|\tau - np| > b\} \le \varepsilon/2.$$

Тогда $P(\tau > \rho) = P\{|\tau - np| > b\}$. Это проиллюстрировано ниже:



$\tau > nP + b$

$\tau - nP > b$

Отсюда следует $P\{\tau > \rho\} \le P\{|\tau - np| > b\} \le D(\tau) / b^2 = \varepsilon/2$.

Лемма доказана.

Лемма 13.3. Пусть $\rho = [E(\tau) + b]$ и $b = \sqrt{\frac{np(1-p)}{\varepsilon/2}}$, тогда

$$1\backslash n \log |S_\rho(x)| \le H(P) + O(n^{-1\backslash 2}) \text{ при } n \to \infty.$$

Доказательство. Из леммы 13.1 и леммы 13.2 получаем неравенство:

$$|S_\rho(x)| \le 2^{nH(\rho\backslash n)}.$$

Прологарифмируем и поделим на n, разложим логарифм в ряд (это корректно) и оставим главные члены:

$$1\backslash n \log |S_\rho(x)| \le H(\rho\backslash n) = -\rho\backslash n \log \rho\backslash n - (1-\rho\backslash n) \log (1-\rho\backslash n) = -[nP+b]\backslash n \log [nP+b]\backslash n - (1-[nP+b]\backslash n] \log (1-[nP+b]\backslash n) \approx -P \log P - (1-P) \log (1-P) + O(b\backslash n) = H(p) + O(n^{-1\backslash 2}).$$

Лемма доказана.

Введем некоторую функцию, определенную на парах векторов из B^n.

$$f(x,y) = \begin{cases} 0, & d(x,y) > \rho \\ 1, & d(x,y) \le \rho \end{cases} \quad (13.5)$$

Лемма 13.4. Справедливо соотношение:

$$P(x^i) \le \sum_{y \in B^n} P(y \mid x^i) \left[1 - f(y, x^i) + \sum_{j \ne i} f(y, x^j) \right]. \quad (13.6)$$

Доказательство. Здесь $P(x^i)$ – вероятность неправильного декодирования переданного слова $x^i \in B^n$. $P(y|x^i)$ - вероятность того, что при передаче слова x^i на декодер попало слово y. Пусть $Q(y|x^i)$- вероятность того, слово y декодировано в слово x^i. Тогда

$$P(x^i) = \sum_{y \in B^n} P(y \mid x^i) \sum_{j \neq i} Q(y, x^j).$$

Заметим, что слово *y* может быть декодировано в слово x^j, только если оно попало в шар соответствующего радиуса с центром в x^j. Из этого следует, что

$$P(x^i) \leq \sum_{y \in B^n} P(y \mid x^i) \sum_{j \neq i} f(y, x^j).$$

Если x^i декодируется правильно, то *y* попало в шар нужного радиуса с центром в x^i , и тогда $\left[1 - f(y, x^i)\right]$=0. В этом случае в неравенстве (13.6) слева 0, а справа неотрицательная величина. Если x^i декодируется неправильно, то *y* попало в шар нужного радиуса с центром в x^j , j≠i, и тогда, если $\left[1 - f(y, x^i)\right]$=0, то

$$P(x^i) \leq \sum_{y \in B^n} P(y \mid x^i) \sum_{j \neq i} f(y, x^j) \leq \sum_{y \in B^n} P(y \mid x^i) \left[1 + \sum_{j \neq i} f(y, x^j) \right].$$

Если же $\left[1 - f(y, x^i)\right]$=1 (а это может быть, если шары пересекаются), то

$$P(x^i) = \sum_{y \in B^n} P(y \mid x^i) \left[1 + \sum_{j \neq i} f(y, x^j) \right].$$

Лемма доказана.

13.4.2 Доказательство теоремы.

Теорема 13.4 (вторая теорема Шеннона). Для любого R: $0 < R < C(\rho)$, такого что $M=2^{[nR]}$, выполняется соотношение:

$$P^*(M, n, P) \to 0, n \to \infty.$$

Доказательство. Пусть передаем x^i, принято некое *y*, оцениваем ошибку декодирования.

Мы рассматриваем в качестве кодов все возможные подмножества B^n. Т.е. кодовые точки располагаются случайным образом. Точки упали как угодно, на них строим шары радиуса ρ.

Декодируем так: если точка *y* попала в шар радиуса *t* вокруг кодовой точки, то декодируем ее в данную кодовую точку, если в этом шаре нет других точек; если есть другие кодовые точки или *y* не попало ни в один из шаров вокруг кодовой точки, то *y* декодируем по эвристическим правилам. При этом возможна ошибка декодирования.

Вспомним введенную выше функцию f(x,y). Из леммы 13.4 следует, что вероятность того, что *y* находится от x^i на расстоянии, большем ρ, задается формулой:

$$\sum_{y \in B^n} P(y \mid x^i) \left[1 - f(y, x^i)\right].$$

С другой стороны, из леммы 13.2 следует, что эта вероятность $P\{ \tau > \rho\} \leq \varepsilon\backslash 2$.

Таким образом

$$P_c = 1\backslash M \sum P(x^i) = \frac{1}{M} \sum_{x^i \in C} \quad P(x^i) \leq \frac{1}{M} \sum_{x^i \in C} \sum_{y \in B^n} P(y \mid x^i) \left[1 - f(y, x^i) + \sum_{j \neq i} f(y, x^j) \right] \leq$$

$$\le \varepsilon/2 + \frac{1}{M}\sum_{x^i\in C}\sum_{y\in B^n}\sum_{j\ne i} P(y\mid x^i)\,f(y,x^j).$$

Заметим теперь, что E(f(y|x^i)- математическое ожидание того, что y попал в шар с центром в x^i, равно:

$$E(f(y|x^i)=|S_\rho(x)|/2^n .$$

Теперь вспомним неравенство:

$$P^*(M,n,p)\le\frac{1}{C^M_{2^n}}\sum_{x^i\in L}P_c(x^i).$$

Заметим теперь, что минимальное значение случайной величины не превосходит ее математического ожидания:

$$\mathrm{Min}\xi\le E(\xi).$$

Из этих соотношений следует цепочка неравенств.

$$P^*(M,n,p)\le\varepsilon/2+\frac{1}{M}\sum_{x^i\in C}\sum_{y\in B^n}\sum_{j\ne i}E(P(y\mid x^i))E(f(y,x^j))=$$

$$=\varepsilon/2+\frac{1}{M}\sum_{x^i\in C}\sum_{y\in B^n}\sum_{j\ne i}E(P(y\mid x^i))\frac{|S_\rho|}{2^n}=\varepsilon/2+\frac{|S_\rho|}{2^n}\frac{1}{M}\sum_{x^i\in C}\sum_{y\in B^n}\sum_{j\ne i}E(P(y\mid x^i))=$$

$$=\varepsilon/2+\frac{|S_\rho|}{2^n}\frac{1}{M}\sum_{x^i\in C}\sum_{j\ne i}E(\sum_{y\in B^n}P(y\mid x^i))=\varepsilon/2+\frac{|S_\rho|}{2^n}\frac{M(M-1)}{M}\le\varepsilon/2+\frac{|S_\rho|}{2^n}M.$$

Таким образом

$$P^*(M,n,p)-\varepsilon\backslash 2\le\frac{|S_\rho|}{2^n}M.$$

Вспомним, что $0<R<C(\rho)$, M=$2^{[nR]}$.

Применяем лемму 13.3 и соотношения: [nR] = log M, $M = 2^{Rn}$:

$$1\backslash n\log(P^*(M,n,P)-\varepsilon\backslash 2)\le 1\backslash n\log(|S_\rho|M\backslash 2^n)=1\backslash n\log|S_\rho|2^{Rn-n}\le R-C(\rho)-1+O(n^{-1\backslash 2})+1.$$

Пусть δ= C(P)-R. Тогда имеем при больших n неравенство

$$P^*(M,n,P)<\varepsilon\backslash 2+2^{-\delta n}.$$

Устремляем n→∞ и δ→0. Получаем

$$P^*(M,n,P)\to 0.$$

Теорема доказана.

14. Лекции 14. Линейные коды: спектры, двойственность.

14.1Линейные коды.

Наложим на множество C ограничение, превратим его из подмножества в подпространство B^n.

Это даст возможность использовать результаты линейной алгебры.

Определение. Код $C \in B^n$ называется линейным (групповым), если для любых $\alpha, \beta \in C$: $\alpha + \beta \in C$.

Сразу из определения следует, что групповому коду обязательно принадлежит нулевой вектор.

Заметим, что если код не является двоичным, то понятия линейного и групповых кодов не совпадают.

В случае двоичного кода: B^n – линейное пространство; зададим в нем подмножество векторов $C \in B^n$, и для любых $\alpha, \beta \in C$: $\alpha + \beta \in C$. Отсюда следует, что вместе с любым набором вектров линейному коду принадлежит их линейная комбинация , поэтому C – подпространство B^n.

Пусть $dimC = k$, $|C| = 2^k$. Согласно определению: C – групповой код.

Легко показать, что $d(c) = \min |\alpha|$ ($\alpha \in C$, $\alpha \neq 0$).

Для любого линейного кода существуют 2 матрицы: G – порождающая, H – проверочная.

$$G = \begin{vmatrix} \alpha^1 \\ \alpha^2 \\ \ldots\ldots \end{vmatrix}$$

Порождающая матрица состоит из векторов (записанных в строке покомпонентно) некоторого базиса подпространства С. Для любого $\beta \in C$: β – линейная комбинация строк G. Если $dimC = k$, то G должна содержать k строк.

Проверочная матрица кода С – это такая матрица, что для любого $\alpha \in C$:

$\alpha H^T = 0$.

При построении кода (алгоритмов кодирования и декодирования) С чаще используется матрица H.

Определение. Код $C^* \in B^n$ называется ортогональным C, если для любых α, β: $\alpha \in C^*$, $\beta \in C$ выполняется условие: $\alpha\beta = 0$.

Из курса линейной алгебры известно, что если $dimC = k$, то $dimC^* = n - k$. При этом порождающей матрицей H * кода C^* является проверочная матрица H кода C . (И наоборот.)

Пример. Код с проверкой на четность.

$dimC = n - 1$, $dimH = 1$, $H = (11\ldots1)_{1\times n}$

$$G = \begin{pmatrix} 1 & 1 & 0 & \ldots & 0 \\ 1 & 0 & 1 & \ldots & 0 \\ 1 & . & . & \ldots & . \\ . & . & . & . & . \\ 1 & 0 & 0 & \ldots & 1 \end{pmatrix}_{(n-1)\times n}$$

Рассмотрим теперь операции преобразования кодовых матриц.

Элементарные преобразования матриц:

1) Перемена местами строк.
2) Умножение строки на число.
3) Сложение строки с линейной комбинацией других строк.
4) Перемена столбцов местами.

Определение. Коды C и D называются эквивалентными, если $G(C)$ можно получить из $G(D)$ с помощью элементарных преобразований.

Утверждение. Для любого C существует эквивалентный код C' такой, что

$$G(C') = [IP] = \left(\begin{array}{ccc|c} 1 & 0 & \ldots & \\ 0 & 1 & \ldots & P \\ . & . & . & \\ 0 & \ldots & 1 & \end{array}\right)$$

$I_{k\times k}$ (k столбцов), $P_{k\times(n-k)}$

$$H(C') = [-P^T I] = \left(\begin{array}{c|ccc} & 1 & 0 & \ldots & 0 \\ P^T & 0 & 1 & \ldots & 0 \\ & . & . & . & . \\ & 0 & 0 & \ldots & 1 \end{array}\right) \quad n-k \qquad (14.1)$$

($n-k$ столбцов)

Определение. Код с порождающей матрицей вида (14.1) называется систематическим.

В таком коде в подматрице $I_{k\times k}$ находятся информационные столбцы, в $P_{k\times(n-k)}$ - проверочные столбцы.

Проверочных символов $(n - k)$. В $H(C')$ $(n - k)$ строк, каждая строка – соотношение, задающее линейную комбинацию: 1-я – 10…0, 2-я – 01…0 и т.д.

Пример.

$n = 5$, $k = 3$ и

$$C = \{\overset{\alpha^1+\alpha^1}{00000}, \overset{\alpha^1}{10011}, \overset{\alpha^2}{01010}, \overset{\alpha^1+\alpha^2}{11001}, \overset{\alpha^3}{00101}, \overset{\alpha^1+\alpha^3}{10110}, \overset{\alpha^1+\alpha^2+\alpha^3}{11100}, \overset{\alpha^2+\alpha^3}{01111}\}$$

$$G = \left|\begin{array}{ccc:cc} \alpha^1 & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{array}\right| \begin{array}{l} \alpha^1 \\ \alpha^2 \\ \alpha^3 \end{array} \qquad H = \left|\begin{array}{ccccc} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{array}\right|$$

P

Пусть α_1,α_2, α_3 – информационные символы, α_4, α_5 – проверочные. Тогда проверочные соотношения:

$$VH^T = 0 \Rightarrow \begin{cases} \alpha_4 = \alpha_1 + \alpha_2 \\ \alpha_5 = \alpha_1 + \alpha_3 \end{cases}$$

Алгоритм кодирования осуществляется следующим образом: любая строка H дает проверочную комбинацию. Так мы преобразуем слова из B^k, подлежащие передаче, просто добавляя к ним n-k символов – результаты проверочных соотношений после подстановки туда информационных символов.

Заметим, что в нашем примере мы получили код с $d(C) = 2$.

В общем случае для декодирования с помощью таблицы декодирования на декодере хранится и в процессе алгоритма декодирования обрабатывается таблица, содержащая $n2^n$ символов (2^n векторов). Вы уже знаете, насколько сложно работать с экспоненциальным объемом информации.

Свойство линейности кода позволяет в этом вопросе получить существенную выгоду. Для хранения таблица декодирования нужно хранить ее первую строку и первый столбец, т.е. $2^{n-k}+2^k$ векторов. Это следует из того, что линейный код B^k – это подгруппа абелевой группы B^n .

Раскладывая группу по подгруппе мы получаем таблицу декодирования в виде матрицы, первая строка которой, как обычно, векторы кода, а остальные – смежные класс. Тогда первый столбец – это образующие смежных классов.

Для линейных кодов такой алгоритм декодирования называется декодированием с помощью «стандартной расстановки».

Определение. Синдром линейного кода называется вектор $S_\alpha = \alpha H^T$.

Если проверочная матрица имеет $(n - k)$ строк, то синдром S_y произвольного вектора y из B^n является вектором длины $(n - k)$.

Поскольку по определению линейного кода вектор y является кодовым тогда и только тогда, когда $yH^T = 0$**,** то справедливо следующее утверждение.

Утверждение. Синдром S_y вектора y равен 0 тогда и только тогда, когда y является кодовым вектором.

Утверждение. Для двоичного линейного кода синдром S_y принятого вектора y равен сумме тех столбцов проверочной матрицы H , где произошли ошибки.

Утверждение. Пусть U – кодовое слово, при передаче из него получено слово V=U+ e, где вектор e – вектор ошибки. Алгоритм на основе стандартной расстановки правильно декодирует слово тогда и только тогда, когда вектор ошибки – образующая смежного класса.

Оно вытекает из следующего утверждения.

Утверждение. Два вектора u и v принадлежат одному и тому же смежному классу тогда и только тогда, когда их синдромы равны.

Достаточно важным для линейных кодов является следующее простое утверждение.

Теорема 14.1. Кодовое расстояние линейного кода $d(C) = d$, тогда и только тогда, когда любые $(d-1)$ столбцов H линейно-независимы и данное d является максимально возможным.

14.2Спектр кода.

Существует еще одна важная характеристика кода – это *спектр* кода.

Рассмотрим код $V = (v_1 \ldots v_k)$. Пусть для кодового слова v_i величина $r_s(v_i)$ – это число кодовых слов таких, что $\rho(v_j, v_i) = s$. Тогда вектору v_i сопоставляется $(r_0(v_i) \ldots r_k(v_i))$ – спектр v_i – кода. А спектры всех кодовых точек можно задавать в виде таблицы размером (k+1)xk.

Если V – линейный код, то спектры всех v_i одинаковые. Поэтому для линейного кода спектром называется вектор $(a_0 \ldots a_n)$, т.е. число кодовых слов весов $0 \ldots n$.

Важным инструментом использования спектра является теорема Маквильямс.

Теорема 14.2. (Маквильямс).

Пусть есть код V и его спектр $(a_0 \ldots a_n)$ и код ортогональный V^* и его спектр $(b_0 \ldots b_n)$, тогда справедливо соотношение $\sum_{i=0}^{n} b_i (1+z)^{n-i}(1-z)^i = 2^{n-k}\sum_{i=0}^{n} a_i z^i$, где z – переменная.

Мы здесь даем саму теорему без доказательства.

Эта теорема дает возможность выражать спектр кода через спектр ортогонального кода.

Оно имеет несколько важных следствий, применяющихся не только в теории кодирования, но в комбинаторике. Приведем некоторые из них.

Следствие1. Пусть $\phi_s(n,i) = \sum_{j=0}^{m} (-1)^j C_i^j C_{n-i}^{s-j}$. Тогда справедливо соотношение

$$a_s = \frac{1}{2^{n-k}} \sum_{i=0}^{n} b_i \varphi_S(n,i).$$

Доказательство. Так как $\phi_s(n,i) = \sum_{j=0}^{m} (-1)^j C_i^j C_{n-i}^{s-j}$, то из $(1+z)^{n-i}(1-z)^i = \sum_{s=0}^{n} \phi_s(n,i) z^s$

с использованием теоремы Маквильмс получаем

$$\sum_{i=0}^{n} b_i (1+z)^{n-i}(1-z)^i = \sum_{i=0}^{n} b_i \sum_{s=0}^{n} \varphi_s(n,i) z^s = \sum_{s=0}^{n} z^s \sum_{i=0}^{n} b_i \varphi_s(n,i) = 2^{n-k} \sum_{s=0}^{n} a_s z^s .$$

Приравнивая коэффициенты при одинаковых степенях, имеем

$$a_s = \frac{1}{2^{n-k}} \sum_{i=0}^{n} b_i \varphi_S(n,i).$$

Следствие доказано.

Заметим, что из $\phi_s(n,i) = \sum_{j=0}^{m} (-1)^j C_i^j C_{n-i}^{s-j}$ следует, что

$$\begin{cases} \varphi_1(n-1,-1) = 2^{n-k} \\ \varphi_1(n-1,i-1) = n-2i+1 \end{cases}.$$

Следствие 2. Справедливо соотношение

$\sum_{i=0}^{n} b_i \varphi_S(n,i)(1+z)^{n-i}(1-z)^i = 2^{n-k} \sum_{r=0}^{n} C(r,s) z^r$, где $C(r,s)$ - число точек B_n веса s на расстоянии r от некоторой точки кода.

Пусть C(s) – число точек B^n веса s, попавших в шар радиуса t линейного кода, исправляющего t ошибок.

Следствие 3. Справедливо соотношение:

$$\sum_{i=0}^{n} b_i \varphi_t(n-i,i-1)(1+z)^{n-i}(1-z)^i = 2^{n-k} \sum_{S=0}^{n} C(s) z^S .$$

Ниже мы приведем пример использования этих соотношений.

14.3 Упражнения.

1. Пусть G – произвольный (n,k)-код, точки которого выписаны в матрицу A размером $(2^k \times n)$. Показать, что каждый ненулевой столбец A содержит ровно 2^{k-1} единиц.

2. Пусть G – произвольный (n,k)-код. Доказать следующие неравенства:

a) $d(G) \le n-k+1$;

b) $d(G) \le \frac{n}{2} \times \frac{2^k}{2^k-1}$.

3. Доказать следующие «общие» свойства функции $A(n,d)$:

$A(n,2t+1) = A(n+1,2t+1)$;

$A(n,d) \le 2A(n-1,d)$.

4*. Доказать, что при n≥2d-1 число проверочных символов, необходимое для того, чтобы минимальный вес линейного кода с n символами достигал значения d, равно, самое меньшее, 2d-*log*d-2.

15. Лекция 15. Дополнения. Примеры.

15.1 Код Хэмминга. Совершенные коды.

Код Хэмминга в то же время является уникальным по структуре объектом, обладающим несколькими интересными свойствами. Это *линейный* код, это *совершенный* код, Код, дуальный к коду Хэмминга, является *эквидистантным* кодом.

Описать код – это описать множество С из 2^n. Код Хэмминга принято задавать через его проверочную матрицу.

Пусть m – натуральное число. Проверочная матрица кода имеет размеры m x (2^m-1).

$$H = \underset{1}{\begin{vmatrix} 1 & 0 & 1 & & \\ 0 & 1 & 1 & & \\ \dots & \dots & \dots & & \\ & \underset{2}{} & \underset{3}{} & & \underset{2^m-1}{} \end{vmatrix}}_{m}$$

В столбцах находятся все ненулевые последовательности длины m, представляющие собой двоичную запись числа – номера столбца. Очевидно, что любые два столбца линейно независимы, а среди троек столбцов уже есть линейно зависимые наборы.

Поэтому, согласно приведенной выше теореме 14.1, кодовое расстояние кода Хемминга d=3. Следовательно, этот код исправляет ровно одну ошибку.

Сразу из определения проверочной матрицы кода Хэмминга следует простой алгоритм декодирования. Декодирование происходит сразу после вычисления синдрома принятого слова y. Для принятого слова y есть три возможности:

1. Ошибок не было. Тогда синдром S = y H^T=0.
2. Ошибка ровно одна. Тогда синдром является двоичной записью (индикатором) номера бита, в котором произошла ошибка, так как для линейного кода сидром – сумма столбцов проверочной матрицы, в которых произошли ошибки. В нашем случае ошибка одна, а столбец – номера столбца
3. Произошло более одной ошибки. Тогда код Хэмминга не обеспечивает защиты от ошибок.

Пример.

m = 3, n = 2^3 – 1 = 7, k = 4. Пусть информационными будут символы ($\alpha_3, \alpha_5, \alpha_6, \alpha_7$).

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

$$\begin{cases} \alpha_1 + \alpha_3 + \alpha_5 + \alpha_7 = 0 \\ \alpha_2 + \alpha_3 + \alpha_6 + \alpha_7 = 0 \\ \alpha_4 + \alpha_5 + \alpha_6 + \alpha_7 = 0 \end{cases} \Leftrightarrow \begin{cases} \alpha_1 = \alpha_3 + \alpha_5 + \alpha_7 \\ \alpha_2 = \alpha_3 + \alpha_6 + \alpha_7 \\ \alpha_4 = \alpha_5 + \alpha_6 + \alpha_7 \end{cases}$$

$U = 1100 \to 0\ 1\ 1\ 1\ 1\ 0\ 0$

Если U – передано, U’ – получено, рассмотрим 3 случая:

1. *Ошибок нет*, $U' = U \Rightarrow UH^T = 0$

2. 1 *ошибка* (*искажен 5й символ*), $U' = 0111000,\ S = \begin{pmatrix}1\\0\\1\end{pmatrix} \Rightarrow 5$ (*позиция, в кот. ош.*)

3. 2 *ошибки* (*искажены 1й и 5й символы*), $U' = 1111000,\ S = \begin{pmatrix}0\\0\\1\end{pmatrix} \Rightarrow 4$ (*неправ. декодирование*)

Заметим, что в данном примере $\begin{pmatrix}0\\0\\1\end{pmatrix} = \begin{pmatrix}1\\0\\0\end{pmatrix} + \begin{pmatrix}1\\0\\1\end{pmatrix}$.

Рассмотрим теперь свойства кода.

Длина кодовых слов $n = 2^m - 1$. Число проверочных символов $m = \log(n + 1)$. Число информационных символов равно $2^m - 1$ - m. Отсюда следует, что число кодовых слов $|C|$ в коде Хэмминга $|C| = 2^n / (1 + n)$.

Заметим теперь, что число точек в шаре радиуса единица равно $1 + n$, и вспомним приведенную выше границу Хэмминга:

$$A(n, 2t+1) \le \frac{2^n}{\sum_{i=0}^{t} C_n^i}.$$

Код Хемминга исправляет одну ошибку и $A(n, 3) = 2 / (1 + n)$. Получаем, что для Хемминга граница Хемминга достигается, шары вокруг кодовых точек покрывают все пространство B^n.

Определение. Код C, исправляющий t ошибок, называется совершенным, если $|C| = \frac{2^n}{\sum_{i=0}^{t} C_n^i}$

.

То есть, совершенный код – код, на котором достигается граница Хемминга.

Таким образом, мы доказали утверждение.

Теорема 15.1. Код Хэмминга является совершенным кодом.

Применим теперь теорему Маквильямс.

Так как код Хэмминга - совершенный код, то число кодовых точек веса s, попавших хотя бы в один шар радиуса единица с центром в кодовой точке, равно общему числу точек веса s, т.е. $C(s) = C^S_n$.

Применяя Следствие 3 из теоремы Маквильямс, получаем

$$\sum_{i=0}^{n} b_i \varphi_1(n-1, i-1)(1+z)^{n-i}(1-z)^i = 2^{n-k}\sum_{S=0}^{n} C_n^S z^S = 2^{n-k}(1+z)^n .$$

Но $\varphi_1(n-1,-1)=2^{n-k}$. Тогда

$$2^{n-k}(1+z)^n + \sum_{i=0}^{n} b_i \phi_1(n-1, i-1)(1+z)^{n-i}(1-z)^i = 2^{n-k}(1+z)^n$$, т.е.

$$\sum_{i=0}^{n} b_i \phi_1(n-1, i-1)(1+z)^{n-i}(1-z)^i = 0 .$$

Т.к. многочлен равен нулю, то все его коэффициенты равны нулю:

$$b_i \phi_1(n-1, i-1) = 0, i = 1, \ldots, n.$$

Но $\phi_1(n-1, i-1) = n - 2i + 1$. Отсюда следует, что $\phi_1(n-1, i-1) = 0$ только при i=(n+1)/2, т.е.

$$b_1 = b_2 = \ldots = b_{(n-1)/2} = b_{(n+3)/2} = \ldots = b_n = 0.$$

Тогда: $n - 2i + 1 = 0$, а $i = (n + 1) / 2$.

Т.е. $b_{(n-1)/2}$ не равно 0, а значит спектр кода V^* равен (0, 0…0, $(n + 1) / 2$, 0…0).

Заметим, кроме того, что справедливо равенство:

$$(n+1)/2=2^{m-1}.$$

Получаем интересный код, который принадлежит к определенному классу.

То есть код, двойственный к коду Хэмминга обладает тем свойство, что попарные расстояния между его кодовыми точками одинаковы и равны (n+1)/2.

15.2 Эквидистантные коды.

Определение. Код называется эквидстититантным, если для любых i, j $(i \neq j)$: $\rho(v_j, v_i) = d$.

Таким образом, мы доказали следующее утверждение.

Теорема 15.2. Код, двойственный коду Хемминга является эквидистантным кодом.

Этот код называется кодом Макдональда.

Параметры кода Макдональда: $n = 2^m - 1$, $k = m$, $d=(n+1)/2$.

Тогда все $\rho(\ldots) = 2^{m-1}$ (число точек в коде Макдональда 2^m)

Применим теорему Плоткина: $A(n, d) \leq 2d / (2d - n)$ для кода Макдональда и получим:

$2d / (2d - n) = 2*2^{m-1} / 2*2^{m-1} - (2^m - 1) = 2^m$

Отсюда следует, что любой эквидистантный код не может содержать более чем 2^m точек. Поэтому код Макдональда оптимален, лучше ничего не построить.

Оказывается «мощностное свойство» кода Макдональда справедливо для любых эквидистантных кодов.

Теорема 15.3. Пусть V – эквидистантный линейный код в B^n, тогда $|V| \le n + 1$.

Доказательство. Пусть $V = (V^1,\ldots,V^s)$, и при этом $\rho(V_j, V_i) = d$, если $i \neq j$.

Тогда имеем

$$\rho(\alpha,\beta) = \sum_{i=1}^{n} |\alpha_i - \beta_i| = \sum_{i=1}^{n} (\alpha_i - \beta_i)^2 = |\overset{=d}{\alpha}| + |\overset{=d}{\beta}| - 2\alpha\beta = d \Rightarrow$$

$$\Rightarrow \alpha\beta = \frac{d}{2}, \quad \alpha \neq \beta.$$

Будем считать, что $0 \subset V$ и пусть это будет первый вектор кода, т.е.

$$V = (\overset{=V}{0}, V^2, \ldots, V^S).$$

Определитель Грама для векторов $V^2,\ldots,V^S$:

$$\Gamma = \begin{vmatrix} d & \frac{d}{2} & \ldots & \frac{d}{2} \\ \frac{d}{2} & d & \ldots & \ldots \\ \ldots & \ldots & . & \ldots \\ \frac{d}{2} & \ldots & \ldots & d \end{vmatrix} = \frac{d^{S-1}S}{2^{S-1}} \neq 0 \Rightarrow V^2,\ldots,V^S$$ - линейно независимы.

Итак, имеем S-1 линейно независимых векторов.

В пространстве B^n может быть не более n линейно независимых векторов $\Rightarrow$

$\Rightarrow S \le n+1$.

Теорема доказана.

Грубо говоря, требование симметрии типа эквидистантности приводит к тому, что в коде остается всего порядка log($|B^n|$) точек.

15.4 БЧХ – коды.

15.4.1 Основные идеи.

В начале 1950-х годов в теории кодирования была поставлена задача построить возможное обобщение кода Хэмминга - код, исправляющий t ошибок. Для этого потребовалось порядка 10 лет.

С этими кодами вы уже знакомы по курсу алгебры. Так что приведенная ниже иллюстрация и обсуждения даны просто в качестве «дополнения» к нашему курсу.

Стояла задача построить (n,k)- код с минимальным числом строк проверочной матрицы и заданным кодовым расстоянием d. Обозначим это минимальное число строк такого кода через m(n,d).

В основу легли идеи:

1. Построить такую проверочную матрицу, у которой кодовое расстояние $2t + 1$ ($2t$ столбцов линейно-независимы).
2. Используя алгебраические свойства кода, построить алгоритм кодирования (декодирования).
3. Определитель Вандермонда

$$\begin{vmatrix} \alpha_1 & & \alpha_n \\ \alpha_1^{\,2} & \dots & \alpha_n^{\,2} \\ \dots & \dots & \dots \\ \alpha_1^{\,d-1} & \dots & \alpha_n^{\,d-1} \end{vmatrix} \quad (d \le n)$$

для матрицы, состоящей из различных элементов поля ($\alpha_i \neq \alpha_j$; i,j=1,…,n) не равен нулю, поэтому любые d-1 столбцов этой матрицы линейно независимы.

Рассмотрим построение проверочной матрицы. Пусть l_1, l_2, …, l_t - локаторы ошибок – двоичные записи номеров битов, в которых произошли ошибки. Тогда проверочная матрица строится так, что ее строки задают проверочные отношения по следующей схеме:

$$H = \begin{vmatrix} & \\ & \\ & \end{vmatrix} \begin{vmatrix} l_1 + l_2 + \dots + l_t \\ l_1^{\,3} + l_2^{\,3} + \dots + l_t^{\,3} \\ l_1^{\,5} + l_2^{\,5} + \dots + l_t^{\,5} \\ \dots \\ l_1^{\,2t-1} + l_2^{\,2t-1} + \dots + l_t^{\,2t-1} \end{vmatrix}$$

Получаем t уравнений с t неизвестными, при декодировании решаем их и находим, где произошли ошибки.

15.4.2 Пример кода, исправляющего две ошибки.

Опишем на следующем примере основную идею декодирования.

Пример. Рассмотрим БЧХ-код, исправляющий две ошибки.

Его проверочная матрица состоит из двух половин: первая половина (верхняя часть матрицы) – та же, что и у кода Хэмминга. Предположим, что множество вектор - столбцов верхней половины – элементы *поля*. (Т.е. их можно складывать, вычитать, умножать и делить.) Тогда вторая половина проверочной матрицы – значения некоторой функции f(i) от векторов первой половин, например, столбцы первой, возведенные в куб (нельзя брать квадраты, потому что квадрат суммы в некоторых полях равен сумме квадратов).

В этом случае первая половина γ вектора синдрома задает, как и раньше, сумму позиций i и j, в которых произошла ошибка, а вторая половина - δ – сумму кубов этих позиций. Если произошло две ошибки, то мы получаем два уравнения с двумя неизвестными. Решаем их (а в поле это можно сделать!) и находим позиции ошибок.

Сказанное проиллюстрировано ниже.

$$\overline{m}\begin{bmatrix} \{B_1 & \ldots\ldots & B_{2^m-1}\} \\ \ldots\ldots & \ldots\ldots & \ldots\ldots \\ m\{f(B_1) & \ldots\ldots & f(B_{2^m-1})\} \end{bmatrix} = H$$

$$V \Rightarrow v + e_i + e_j$$

$$\left.\begin{array}{l} vH^T = 0 \\ (v + e_i + e_j)\cdot H^T \neq 0 \end{array}\right\} \Rightarrow \begin{cases} i + j \Rightarrow \textit{сумма испорченных столбцов} \\ f(i) + f(j) \Rightarrow \textit{получается умножением на нижнюю} \cdot \textit{пол. } H \end{cases}$$

Мы имеем систему двух уравнений с двумя неизвестными:

$$\begin{cases} i + j = \gamma \\ i^3 + j^3 = \delta \end{cases}$$

В поле GF(2^n) (см. Замечание ниже) можно получить:

$$\gamma^2 = (i + j)(i^2 + ij + ij + j^2) = i^2 + j^2.$$

Отсюда

$$\delta = (i + j)(i^2 - ij + j^2) = \gamma(i^2 + ij + j^2) = \gamma(ij - \gamma^2).$$

Тогда удовлетворяют уравнению $x^2 + \gamma x + \gamma^2 + \delta/\gamma = 0$.

Это уравнение решается в поле GF(2^n) и всегда имеет там два корня.

Замечание.

На практике вектора i и j называются векторами локаторов ошибок (или просто локаторами ошибок), а многочлен, корнями которого являются вектора, мультипликативно обратные локаторам ошибок, называется многочленом локаторов ошибок. И именно он используется в алгоритмах кодирования и декодирования, вместо приведенного выше.

В общем случае это следующий многочлен:

$$\sigma(z) = \prod_x (1 - xz),$$

где произведение берется по всем локаторам ошибок.

Для нашего примера о н выглядит так: $x^{-2}(\gamma^2 + \delta/\gamma) + \gamma x^{-1} + 1 = 0$.

15.4.3 Основная теорема.

Пусть $\alpha_1 \ldots \alpha_n$ – элементы поля $GF(2^k)$, а $\gamma_1, \gamma_2, \ldots, \gamma_i$- соответствующие им двоичные векторы, записанные в виде столбцов. Эьл элементы поля, которое вы знаете, как строить.

Рассмотрим матрицу.

$$M = \begin{vmatrix} \gamma_1 & & \gamma_n \\ \gamma_1^3 & \dots & \gamma_n^3 \\ \dots & \dots & \dots \\ \gamma_1^{2S-1} & \dots & \gamma_n^{2S-1} \end{vmatrix}_{kn \times n}$$ - (0,1)-матрица размера knxn.

Теорема 15.4. Любые 2*S* столбцов матрицы M линейно-независимы.

Доказательство. От противного. (Всюду «+» означает сложение по *mod 2*.

Заметим, что

$$(\alpha_1 + \dots + \alpha_n)^2 = \alpha_1^2 + \dots + \alpha_n^2 \qquad (15.1)$$

Пусть существует h столбцов ($h \le 2S$), являющихся линейно-зависимыми. Это столбцы с номерами $i_1 \dots i_h$, тогда

$\gamma^{2t-1}{}_{i1} + \dots + \gamma^{2t-1}{}_{ih} = (0 \dots 0)^T$, $t = 1 \dots S$

Если перейти к элементам поля получим:

$$\alpha^{2t-1}{}_{i1} + \dots + \alpha^{2t-1}{}_{ih} = 0,\ t = 1 \dots S \qquad (15.2)$$

Для любого четного q ($1 \le q \le 2S$): можно представить $q = 2^U(2t - 1)$, $U \ge 1$, $t \le S$

Применим соотношение (15.1) к равенству (15.2) U раз и получим:

$$\alpha^{q}{}_{i1} + \dots + \alpha^{q}{}_{in} = 0. \quad (15.3)$$

Таким образом

$\alpha^{j}{}_{i1} + \dots + \alpha^{j}{}_{in} = 0$ при j=1,…,2S.

Возьмем матрицу M, определитель любого минора порядка $(d - 1)$ будет представлять собой определитель Вандермонда порядка $d - 1$. А он не равен нулю.

Теперь возьмем $d = 2S + 1$, т.о. любой минор порядка 2*S* не равен 0. Отсюда следует, что любые 2*S* столбцов линейно-независимы.

Получили противоречие с соотношением (15.3).

Теорема доказана.

Следствие. Матрица M может быть проверочной матрицей (n, k) – кода, исправляющего *S* ошибок.

Примеры.

Код Хэмминга является БЧХ-кодом с кодовым расстояние 3 и исправляет одну ошибку.

В приведенном выше примере проверочная матрица кода с расстоянием 5 строится из двух частей. Верхняя часть – проверочная матрица кода Хэмминга. Нижняя половина – столбцы верхней половины, возведенные в куб (эти столбцы трактуются как элементы описанного выше поля. Схема декодирования приведена выше.

Для случая трех ошибок проверочная матрица состоит уже из трех частей. Третья часть – это столбцы первой части, возведенные в пятую степень. Многочлен денумераторов ошибок имеет третью степень. Пусть произошло 3 ошибки в позициях η, δ, ζ. Тогда при

рассмотрении верхней части матрицы $\eta + \delta + \zeta = \rho^{*}$; второй части: $\eta^3 + \delta^3 + \zeta^3 = \rho^{**}$; третьей части: $\eta^5 + \delta^5 + \zeta^5 = \rho^{***}$. Получаем систему из трех уравнений с тремя неизвестными.

Теория и алгоритмы решения уравнений в конечных полях хорошо разработаны, поэтому решение уравнений в алгоритме декодирования БЧХ-кода не представляет труда. Конечно, чем больше *S*, тем больше сложность алгоритма.

Число проверочных символов БЧХ-кода для исправления *S* ошибок (упомянутое выше минимальное число строк проверочной матрицы) подчиняется соотношению:

$m(n,d) \sim s \log n$

$k = n - s \log n$ при $n \to \infty$, $k \to n$.

Утверждение. Для любой бесконечной последовательности (n,k,d)-кодов БЧХ над GF (q) скорость кода k/n и отношение d/n стремятся к нулю с ростом n.

Определение. Линейный код длины *n* называется *циклическим,* если для любого кодового слова $(x_1, x_2, \ldots, x_n)$ слово $(x_2, \ldots, x_n, x_1)$ также является кодовым.

В курсе алгебры вам было доказано, что БЧХ-код является циклическим кодом.

Циклические коды обладают несложными процедурами кодирования и декодирования, имеют хорошие алгебраические свойства, их группы автоморфизмов содержат циклические подстановки. Все совершенные линейные коды, коды Рида-Маллера и другие коды являются кодами, которые после перестановки координат и несущественной модификации оказываются циклическими. Многие важнейшие блоковые коды, открытые позже, также оказались либо циклическими, либо тесно связанными с ними.

Построение БЧХ-кодом явилось качественным прорывом не только в такой математической дисциплине как *теория кодирования,* но и в инженерных технологиях защиты информации от шума в канале.

Выше мы уже упоминали о том, что в протоколах передачи информации встают вопросы синхронизации скорости генерации кодовых слов, скорости передачи и пропускной способности канала.

Таким же важным вопросам для адаптивных протоколов является подбор соответствия корректирующей способности кода и качества канала. Очевидно, что при вероятности ошибки в канале p=0,2 и длине кодового слова n=10 использование кода с d=4 или даже 5 означает сознательное согласие на то, что информация будет частично искажаться. Если же кодовое расстояние будет, например, 20, то вероятность искажения информации будет меньше, чем 10^{-10}.

Отсюда следует, что при построении протокола передачи необходимо учитывать такие свойства информации, как *важность*, *ценность*, *чувствительность* к *искажению*. Для их учета нужно уметь их измерять (*кодировать в широком смысле*!). Но для этого нужно дать им определение.

Как ни странно, на сегодня этот вопрос еще малоизучен. В протоколах передачи в лучшем случае используются эвристики и экспертные оценки. Типичный подход: «Данная информация является важной и требует передачи с гарантией безошибочности $P_\gamma=10^{-9}$.»

А если мы хотим передавать по каналу разные типы информации? На практике для этого используются не только разные каналы, но и разные сети. В то же время, с математической точки зрения, очевидно, что, зная p, n и P_γ, можно подобрать d.

Такого сорта протоколы нельзя построить, имея в своем распоряжении только код, исправляющий одну ошибку. Именно появление кодов с различными значениями параметра d (например, БЧХ-кодов) дало такую возможность.

16. Приложение. Производящие функции и метод коэффициентов.

Метод коэффициентов является одним из вариантов метода производящих функций. При этом используемый формализм в ряде случаев существенно удобнее классического варианта. Обозначим через $C(u)$ — класс формальных степенных рядов, содержащих лишь конечное число членов с отрицательными степенями. Точнее, $C(u) = \{A(u)\}$:

$$A(u) = u^k(a_0 + a_1 u + \cdots), \text{где } a_0 \neq 0 \text{ и } k = 0, \pm 1, \pm 2, \ldots$$

Далее положим по определению

$$\underset{u}{coef}\,\{A(u)\} = a_{-1}$$

Другими словами, на множестве $C(u)$ определён линейный функционал, ставящий каждому степенному ряду в соответствие число — коэффициент при u^{-1}. Следующие свойства этого функционала являются прямыми следствиями определений.

1. $\underset{u}{coef}\,\{\lambda A(u) + \mu B(u)\} = \lambda \underset{u}{coef}\,\{A(u)\} + \mu \underset{u}{coef}\,\{B(u)\}$ — линейность.

2. Если $A(u)$ — «обычный» степенной ряд, не содержащий отрицательных степеней u, то

$$\sum_{k=0}^{\infty} z^k \underset{u}{coef}\,\{A(u)u^{-k-1}\} = A(z)$$

   Свойство 2 называется заменой переменных.

3. Если ряд $A(u)$ сходится в окрестности нуля, то

$$\underset{u}{coef} A(u) = \frac{1}{2\pi i}\oint_{|u|=\rho} A\,(u)du = \underset{u=0}{res}\{A(u)\}$$

Таким образом, для степенных рядов $A(u)$, сходящихся в окрестности нуля, $\underset{u}{coef}\{A(u)\}$ совпадает с вычетом $A(u)$ в $u = 0$.

Следующая таблица является переложением на используемый формализм классических формул из анализа.

$\underset{u}{coef}\,\{(1+u)^n u^{-k-1}\}$	$\binom{n}{k}$
$\underset{u}{coef}\,\{e^{\lambda u}u^{-k-1}\}$	$\frac{\lambda^k}{k!}$
$\underset{u}{coef}\,\{\ln(1-u)\,u^{-k-1}\}$	$-\frac{1}{k}$

$\underset{u}{coef}\{(1+u)^{-n}u^{-k-1}\}$	$\binom{n+k-1}{k}$
$\underset{u}{coef}\left\{(1-4u)^{-\frac{1}{2}}u^{-k-1}\right\}$	$\binom{2k}{k}$
$\underset{u}{coef}\{(1-u)^{n}u^{-k-1}\}$	$(-1)^k\binom{n}{k}$

Метод коэффициентов успешно применяется для вычисления комбинаторных сумм, нивелирования ограничений, описывающих область суммирования, построения композиций, упрощающих суммирование.

Комбинаторные суммы

Формального определения понятия «комбинаторная сумма» не существует, что однако не препятствует широкому использованию этого термина в практических приложениях.

Пример. Рассмотрим следующую сумму:

$$S_n = \sum_{k=0}^{r}(-1)^k\binom{n}{k}.$$

Для вычисления S_n заменим биномиальный коэффициент $\binom{n}{k}$ его табличным выражением и продолжим вычисление суммы S_n, используя правило линейности:

$$S_n = \sum_{k=0}^{r}\underset{u}{coef}\{(1-u)^n u^{-k-1}\} = \underset{u}{coef}\left\{\frac{(1-u)^n}{u}\sum_{k=0}^{r}\left(\frac{1}{u}\right)^k\right\} = = \underset{u}{coef}\left\{\frac{(1-u)^n}{u}\frac{1-\frac{1}{u^{r+1}}}{1-\frac{1}{u}}\right\}$$

$$= \underset{u}{coef}\frac{(1-u)^n}{u-1}\left(1-\frac{1}{u^{r+1}}\right) == -\underset{u}{coef}(1-u)^n + \underset{u}{coef}\frac{(1-u)^{n-1}}{u^{r+1}}$$

$$= \underset{u}{coef}\frac{(1-u)^{n-1}}{u^{r+1}} == (-1)^r\binom{n-1}{r}.$$

Таким образом, окончательный ответ таков:

$$S_n = (-1)^r\binom{n-1}{r}.$$

Пример. Рассмотрим сумму Вандермонда:

$$S_n = \sum_{k=0}\binom{N}{n+k}\binom{M}{m-k}.$$

Здесь при суммировании используются обычные правила действия с биномиальными коэффициентами и следующие соглашения:

$$\binom{n}{k} = \begin{cases}0, & \text{при } k > n \text{ и } k < 0,\\ 1, & \text{при } k = 0.\end{cases}$$

Опять используя таблицу замен биномиальных коэффициентов, получаем выражения:

$$S=\sum_{k=0}\binom{N}{n+k}\binom{M}{m-k}=\sum_{k=0}\underset{u}{coef}\left\{(1+u)^N u^{-n-k-1}\right\}\underset{v}{coef}\left\{(1+v)^M v^{-m+k-1}\right\}=$$

$$=\underset{v}{coef}\left\{\frac{(1+v)^M}{v^{m+1}}\sum_{k=0}v^k\underset{u}{coef}\left\{\frac{(1+u)^N}{u^n}u^{-k-1}\right\}\right\}=$$

$$=\underset{v}{coef}\left\{\frac{(1+v)^M}{v^{m+1}}\frac{(1+v)^N}{v^n}\right\}=\underset{v}{coef}\left\{\frac{(1+v)^{M+N}}{v^{m+n+1}}\right\}=\binom{M+N}{m+n}$$

17. Благодарность.

Авторы выражают глубокую признательность студенту ФУПМ Сергееву Федору Игоревичу за большую помощь при наборе данного текста.

18. Рекомендованная литература

1. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.
2. Лидовский В.В. Теория информации: Уч. пособие. - М.: Компания Спутник+, 2004.
3. Потапов В.Н. Теория информации. Кодирование вероятностных источников.- Уч. пособие / Новосибирский государственный университет. -- Новосибирск, 1999.
4. Леонтьев В.К. Теория кодирования.- М., Знание, 1977.
5. Леонтьев В.К. Комбинаторика и информация. Часть 1: Комбинаторный анализ.- М., МФТИ, 2015.
6. Леонтьев В.К. Комбинаторика и информация. Часть 2: Информационные модели. - М., МФТИ, 2015.
7. У. Питерсон, Э. Уэлдон. Коды, исправляющие ошибки. – Мир, 1976.
8. Соловьева Ф.И. Введение в теорию кодирования. Уч. Пособие. – Новосибирск, 2006.
9. Зуев Ю.А. По океану дискретной математики. Т.1,2. –М.: Либроком, 2012.
10. Грехем Р., Кнут Д., Паташник О. Конкретная математика.–М.: Мир, 1998.
11. Мак-Вильямс Ф. Дж., Слоэн Н. Дж. Теория кодов, исправляющих ошибки. –М.: Связь, 1979.
12. Леонтьев В.К. Избранные задачи комбинаторного анализа. –М.: Из-во МГТУ им. Н.Э. Баумана, 2001.
13. Егорычев Г.П. Интегральное представление и вычисление комбинаторных сумм. – Новосибирск: Наука, 1977.
14. Leont'ev V., Smetanin Yu.G. Problems of Information on the Set of Words // J.Math.Sci. 2002. V.108. №1. P. 49-71