МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)

Е.В. ГОШИН

ТЕОРИЯ ИНФОРМАЦИИ И КОДИРОВАНИЯ

Рекомендовано редакционно-издательским советом федерального государственного автономного образовательного учреждения высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева» в качестве учебного пособия для студентов, обучающихся по основной образовательной программе высшего образования по направлению подготовки 01.03.02 Прикладная математика и информатика и специальности 10.05.03 Информационная безопасность автоматизированных систем

САМАРА
Издательство Самарского университета
2018

УДК 519.72(075)
ББК 32.811я7
Г749
Рецензенты: д-р техн. наук С. А. П р о х о р о в,
д-р техн. наук С. Б. П о п о в

Гошин, Егор Вячеславович
Г749 **Теория информации и кодирования:** учеб. пособие. – Самара: Изд-во Самарского университета, 2018. – 124 с.

ISBN 978-5-7883-1260-6

В учебном пособии рассматриваются основы теории информации и кодирования. В качестве теоретической основы приведены понятия энтропии и количества информации. Рассмотрены основные типы дискретных и непрерывных каналов, для них приведены и обоснованы численные характеристики пропускной способности. Значительная часть пособия посвящена методам и алгоритмам кодирования источника и кодирования канала. В частности, рассмотрены следующие подходы: кодирование Шеннона-Фано; кодирование Хаффмена; арифметическое кодирование; циклические коды, в том числе, исправляющие пакеты ошибок; алгоритм подсчёта контрольной суммы; коды Адамара; коды Рида-Маллера; свёрточные коды.

Предназначено для студентов, обучающихся по направлению подготовки 01.03.02 Прикладная математика и информатика и специальности 10.05.03 Информационная безопасность автоматизированных систем.

УДК 519.72(075)
ББК 32.811я7

ISBN 978-5-7883-1260-6

ОГЛАВЛЕНИЕ

Предисловие

Автор настоящего пособия читает курс лекций по теории информации в Самарском университете на кафедре суперкомпьютеров и общей информатики. При этом последние несколько лет автор в основном пользовался изданным профессором Фурсовым В.А. в 2013 году учебным пособием «Лекции по теории информации», полностью соответствующим программе курса. Однако, в ходе работы содержание некоторых лекций было переработано, часть лекций – полностью заменены, что и послужило мотивацией для написания настоящего учебного пособия.

В настоящем пособии с небольшими изменениями приведены основы теории информации в части тем, связанных с энтропией и количеством информации. Существенно переработана и дополнена часть, посвящённая дискретным и непрерывным каналам. Кардинально переработаны главы, посвящённые эффективному кодированию; существенно дополнен раздел, посвящённый помехоустойчивому кодированию.

Учебное пособие предназначено в основном для подготовки бакалавров по направлению 01.04.02 Прикладная математика и информатика и специальности

10.05.03 Информационная безопасность автоматизированных систем, но может быть полезно и для студентов других специальностей и направлений.

Автор выражает благодарность профессору Фурсову В.А. за огромный труд по подготовке предыдущего издания и чтению курса лекций «Теория информации», без которого настоящее издание не могло бы появиться на свет.

Рекомендации по распределению тем

В таблице 1 приведены рекомендации по распределению тем настоящего учебного пособия по лекциям в зависимости от числа лекций.

Таблица 1. Распределение тем по лекциям

<table>
<tr><th></th><th>18 лекций</th><th>9 лекций</th></tr>
<tr><td>Введение в теорию информации.</td><td rowspan="2">1</td><td rowspan="3">1-2</td></tr>
<tr><td>Тема 1. Ансамбли и вероятности. Байесовский вывод.</td></tr>
<tr><td>Тема 2. Энтропия.</td><td>2-3</td></tr>
<tr><td>Тема 3. Количество информации</td><td>4</td><td rowspan="2">4</td></tr>
<tr><td>Тема 4. Каналы передачи данных</td><td>5</td></tr>
<tr><td>Тема 5. Символьные коды. Префиксные коды.</td><td>6</td><td rowspan="2">4</td></tr>
<tr><td>Тема 6. Кодирование Шеннона-Фано. Кодирование Хаффмена. Арифметическое кодирование.</td><td>7</td></tr>
<tr><td>Тема 7. Другие эффективные коды.</td><td>8</td><td>4</td></tr>
<tr><td>Тема 8. Помехоустойчивое кодирование. Код Хэмминга.</td><td>9-10</td><td>5</td></tr>
<tr><td>Тема 9. Циклические коды.</td><td>11</td><td rowspan="2">6</td></tr>
<tr><td>Тема 10. Исправление пакетов ошибок. CRC.</td><td>12</td></tr>
<tr><td>Тема 11. Матричные коды. Коды Адамара</td><td>13</td><td>7</td></tr>
<tr><td>Тема 12. Коды Рида-Маллера.</td><td>14</td><td>8</td></tr>
<tr><td>Тема 13. Свёрточные коды. Треллис-диаграммы.</td><td>15</td><td>9</td></tr>
<tr><td>Тема 14. Модели детерминированных сигналов.</td><td rowspan="2">16</td><td rowspan="2">-</td></tr>
<tr><td>Тема 15. Восстановление сигнала по его дискретным значениям.</td></tr>
</table>

Введение в предмет

> «Фундаментальной задачей коммуникации является точное или приблизительное воспроизведение в некоторой точке сообщения, выбранного в другой точке.»
>
> Клод Шеннон, 1948 г.

Теория информации – это раздел прикладной математики, радиотехники (теория обработки сигналов) и информатики, относящийся к измерению количества информации, её свойств и устанавливающий предельные соотношения для систем передачи данных.

Рассмотрим общую схему передачи данных (рисунок 1), и вкратце опишем каждый блок представленной схемы.

Кодирование источника применяется для того, чтобы минимизировать число бит в единицу времени, необходимых для представления выходных данных источника. Этот процесс известен как кодирование источника или сжатие данных.

Примеры: кодирование Хаффмена, алгоритм Лемпеля-Зива.

Шифрование применяется для обеспечения безопасности передачи битов источника. Процесс преобразования исходных битов (сообщения) в поток бессодержательно выглядящих битов (зашифрованный текст) носит название шифрования.

Примеры: стандарт шифрования данных (DES), RSA.

Кодирование канала применяется для коррекции ошибок, вносимых средой передачи данных. Процесс заключается во введение в последовательность нескольких избыточных битов по заданному правилу для коррекции возникающих ошибок.

Пример: коды повторения, коды Хэмминга, коды Рида-Маллера, циклические коды, CRC-коды.

Модуляция – процесс преобразования цифрового сигнала в аналоговый для передачи по физическому каналу.

Примеры: PSK, QAM.

Канал – физическая среда передачи данных. В ходе передачи данные могут быть искажены вследствие различных эффектов: шума, интерференции, затухания сигнала.

Примеры: двоичный канал (со стиранием), канал с добавлением белого шума.

Демодуляция, *декодирование канала*, *дешифрование* и *декодирование источника* представляют собой обратные процедуры для модуляции, кодирования канала, шифрования и кодирования источника, соответственно.

Рис. 1. Схема передачи данных

Актуальными задачами теории информации являются:

- Оценка характеристик источника и канала (энтропия и количество информации).
- Интерпретация данных (байесовский вывод).
- Сжатие данных (эффективное кодирование)
- Коррекция ошибок (помехоустойчивое кодирование)

Тема 1. Ансамбли и вероятности. Байесовский вывод

Ансамбль X – это тройка (x, A_x, P_x), где исход x – это значение некоторой случайной величины, принимающей одно из набора возможных значений $A_x = \{a_1, a_2, ..., a_i, ..., a_I\}$ с вероятностями $P_x = \{p_1, p_2, ..., p_i, ..., p_I\}$.

$P(x = a_i) = p_i$, $p_i \geq 0$, $\sum_{a_i \in A_x} P(x = a_i) = 1$.

Вероятность подмножества.

Если T – подмножество A_x, тогда

$P(T) = P(x \in T) = \sum_{a_i \in T} P(x = a_i)$.

Совместный ансамбль XY – это ансамбль, каждый исход которого представляет собой упорядоченную пару x,y, в которой $x \in A_x = \{a_1, a_2, ..., a_I\}$, а $y \in A_y = \{b_1, b_2, ..., b_J\}$.

Будем называть вероятность $P(x, y)$ *совместной вероятностью* x и y. В такой записи запятая опциональна, поэтому $P(x, y)$ и $P(xy)$ суть одно и то же. Следует обратить внимание, что случайные величины x и y, входящие в ансамбль XY могут не быть независимыми.

Вероятности отдельных величин $P(x)$ и $P(y)$, входящих в ансамбль определяются через совместные вероятности как

$P(x = a_i) = \sum_{b_j \in A_Y} P(x = a_i, y = b_j)$,

$P(y) = \sum_{y \in A_Y} P(x, y)$.

Вероятность того, что x равно a_i при условии, что $y = b_j$ называется условной вероятностью, обозначается и определяется следующим образом:

$$P(x = a_i \mid y = b_j) = \frac{P(x = a_i, y = b_j)}{P(y = b_j)} \text{ при } P(y = b_j) \neq 0$$

Правило умножения:

$$P(x,y\,|\,H)=P(x\,|\,y,H)P(y\,|\,H)=P(y\,|\,x,H)P(x\,|\,H).$$

Правило суммирования:

$$P(x\,|\,H)=\sum_{y}P(x,y\,|\,H)=\sum_{y}P(x\,|\,y,H)P(y\,|\,H).$$

Теорема Байеса

$$P(y\,|\,x,H)=\frac{P(x\,|\,y,H)P(y\,|\,H)}{P(x\,|\,H)}=\frac{P(x\,|\,y,H)P(y\,|\,H)}{\sum_{y'}P(x\,|\,y',H)P(y'\,|\,H)}.$$

Независимость

Две случайные величины X и Y независимы ($X\perp Y$) тогда и только тогда, когда

$$P(x,y)=P(x)P(y).$$

Пример:

Спам-фильтр работает с 95% надёжностью, то есть, сообщения со спамом фильтруются с вероятностью 95%, а сообщения без спама с вероятностью 5%.

В среднем, 25% приходящих сообщений – спам. Какова вероятность, что отфильтрованное сообщение не содержит спама.

Решение:

Пусть a = 1 – сообщение содержит спам, a = 0 – не содержит.

Результат фильтрации b =1 – сообщение отфильтровано, b = 0 – нет.

Тогда

$$P(b=1\,|\,a=1)=0,95, \qquad P(b=1\,|\,a=0)=0,05,$$

$$P(b=0\,|\,a=1)=0,05, \qquad P(b=0\,|\,a=0)=0,95.$$

Априорная вероятность наличия/отсутствия спама

$$P(a=1)=0,25, \qquad P(a=0)=0,75,$$

Какова общая вероятность фильтрации сообщения?

$$P(b=1)=P(b=1\,|\,a=1)P(a=1)+P(b=1\,|\,a=0).$$

По формуле Байеса вероятность, что отфильтрованное сообщение не содержало спам:

$$P(a=0\,|\,b=1)=\frac{P(b=1\,|\,a=0)P(a=0)}{P(b=1\,|\,a=1)P(a=1)+P(b=1\,|\,a=0)}=$$

$$=\frac{0,05\cdot 0,75}{0,95\cdot 0,25+0,05\cdot 0,75}=0,1(36).$$

Математическое ожидание случайной величины.

Дискретный случай:

$$MX=\sum_i x_i p_i\,.$$

Непрерывный случай:

$$MX=\int_{x=-\infty}^{+\infty} xf(x)dx\,.$$

Дисперсия случайной величины.

Дискретный случай:

$$DX=\sum_i (x_i-MX)^2\,p_i\,.$$

Непрерывный случай:

$$DX=\int_{x=-\infty}^{+\infty}(x-MX)^2\,f(x)dx\,,$$

$$DX=MX^2-(MX)^2.$$

Дискретные законы распределения

Бернулли (параметр p) – описывает успех (или провал) в одиночном испытании:

$$p(x)=\begin{cases}p, & k=1,\\ 1-p, & k=0,\end{cases}$$

$$MX=p,$$

$$DX=p(1-p).$$

Биномиальный закон распределения (параметры p и n) – описывает число успехов в n независимых испытаниях Бернулли.

$$p(k)=C_n^k p^k(1-p)^{n-k},\ k=0,...,n,$$

$$MX=np,$$

$$DX=np(1-p).$$

Геометрическое распределение (параметр p) – число попыток до первого успеха

$$p(k) = (1-p)^{k-1} p, \qquad k = 1,...,n,...$$

$$MX = \frac{1}{p},$$

$$DX = \frac{1-p}{p^2}.$$

Непрерывные законы распределения

Равномерный закон распределения на интервале [a,b]

$$f(x) = \begin{cases} 0, & x < a \\ \dfrac{1}{b-a}, & a \le x \le b, \\ 0, & x > b, \end{cases} \qquad MX = \frac{a+b}{2}, \quad DX = \frac{(b-2)^2}{12}.$$

Экспоненциальный:

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & x \ge 0, \\ 0, & x < 0, \end{cases} \qquad MX = \frac{1}{\lambda}, \quad DX = \frac{1}{\lambda^2}.$$

Нормальный:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \qquad MX = \mu, \quad DX = \sigma^2.$$

Байесовский вывод

Формула Байеса:

$$P(\lambda \mid \{x_1,...,x_N\}) = \frac{P(\{x_1,...,x_N\} \mid \lambda) P(\lambda)}{P(\{x_1,...,x_N\})}.$$

Три задачи могут решаться на основе наблюдения $\{x_1,...,x_N\}$:

- оценка распределения источника;
- предсказание результата;
- сравнение гипотез.

Задача 1.

Два человека оставили следы своей крови на месте преступления. По результатам анализа крови у подозреваемого Оливера, была определена первая группа крови. В следах на месте преступления обнаружена кровь была двух групп: первой (самая распространённая, наблюдается у 60% населения) и четвёртой (редкая, 1% населения). Полученные результаты (первая и четвёртая группы) говорят в пользу присутствия Оливера на месте преступления или наоборот?

Решение.

Обозначим D – данные, S – предположение о том, что «подозреваемый и ещё один неизвестный человек присутствовали на месте преступления», а $\overline{S}$ – «два неизвестных человека присутствовали на месте преступления»

$$P\left(D \mid S, H\right) = p_4,$$

$$P\left(D \mid \overline{S}, H\right) = 2p_1 p_4,$$

$$\frac{P\left(D \mid S, H\right)}{P\left(D \mid \overline{S}, H\right)} = \frac{1}{2p_1} = \frac{1}{1{,}2} \approx 0{,}83.$$

Можно решить предыдущую задачу в общем случае для n_1 образцов крови 1 группы и n_4 – 4 группы. Распределение групп крови p_1 и p_4.

$$P\left(n_1, n_4 \mid S\right) = \frac{\left(N-1\right)!}{\left(n_1-1\right)! n_4!} p_1^{n_1-1} p_4^{n_4},$$

$$P\left(n_1, n_4 \mid \overline{S}\right) = \frac{\left(N\right)!}{n_1! n_4!} p_1^{n_1} p_4^{n_4},$$

$$\frac{P\left(n_1, n_4 \mid S\right)}{P\left(n_1, n_4 \mid \overline{S}\right)} = \frac{\dfrac{\left(N-1\right)!}{\left(n_1-1\right)! n_4!} p_1^{n_1-1} p_4^{n_4}}{\dfrac{\left(N\right)!}{n_1! n_4!} p_1^{n_1} p_4^{n_4}} = \frac{n_1 / N}{p_1}.$$

Задача 2.

Погнутая монета бросается F раз. Мы наблюдаем последовательность из s орлов и решек, которые мы обозначим a и b соответственно. Мы хотим знать насколько неравномерны результаты бросков и предсказать вероятность того, что следующий бросок будет орлом.

Предположим равномерное априорное распределение и получим апостериорное распределение посредством умножения на правдоподобие.

Пусть H_1 – наше предположение. При заданной p_a вероятности, что F бросков сформируют последовательность $\mathbf{s}$, содержащую $\{F_a, F_b\}$ орлов и решек равна

$$P(\mathbf{s} \mid p_a, F, H_1) = p_a^{F_a} (1 - p_a)^{F_b}.$$

Например, $P(\mathbf{s} = \{a,a,b,a\} \mid p_a, F = 4, H_1) = p_a p_a (1 - p_a) p_a$. Наша первая модель предполагает равномерное распределение для p_a

$$P(p_a \mid H_1) = 1, \quad p_a \in [0,1],$$

и

$$p_b \equiv 1 - p_a.$$

Оценка неизвестных параметров

Пусть дана строка длины F в которой F_a – число орлов, а F_b – число решек, мы оценим (а) каким может быть p_a, (б) предскажем, будет следующий символ a или b. [Предсказания обычно выражаются в форме вероятностей, поэтому «предсказать, будет ли следующий символ a» то же самое, что вычислить вероятность того, что следующий символ a].

Предположим, что H_1 – истина. апостериорная вероятность p_a, при заданной строке длины F, которая содержит $\{F_a, F_b\}$ орлов и решек по теореме Байеса равна

$$P(p_a \mid \mathbf{s}, F, H_1) = \frac{P(\mathbf{s} \mid p_a, F, H_1) P(p_a \mid H_1)}{P(\mathbf{s} \mid F, H_1)}.$$

Множитель $P(\mathbf{s} \mid p_a, F, H_1)$, являющийся функцией от p_a и известный как функция правдоподобия, и априорная вероятность $P(p_a \mid H_1)$ определены нами ранее. Наша оценка p_a таким образом равна

$$P(p_a \mid \mathbf{s}, F, H_1) = \frac{p_a^{F_a} (1 - p_a)^{F_b}}{P(\mathbf{s} \mid F, H_1)}.$$

Нормализующая константа дана бета-интегралом

$$P(\mathbf{s} \mid F, H_1) = \int_0^1 \mathrm{d}\, p_a^{F_a} (1 - p_a)^{F_b} = \frac{\Gamma(F_a + 1)\Gamma(F_b + 1)}{\Gamma(F_a + F_b + 2)} = \frac{F_a! F_b!}{(F_a + F_b + 1)!}.$$

От оценки к предсказанию.

Наше предположение о следующем броске, вероятность того, что результатом его будет a, получается путём интегрирования по p_a. Это является результатом неопределённости p_a по нашим предположениям. По правилу суммирования

$$P(a \mid \mathbf{s}, F) = \int \mathrm{d}p_a P(a \mid p_a) P(p_a \mid \mathbf{s}, F).$$

Вероятность того, что выпадет а при заданном p_a равна p_a, поэтому

$$P(a \mid \mathbf{s}, F) = \int \mathrm{d}p_a\, p_a \frac{p_a^{F_a} (1 - p_a)^{F_b}}{P(\mathbf{s} \mid F)} =$$

$$= \int \mathrm{d}p_a \frac{p_a^{F_a + 1} (1 - p_a)^{F_b}}{P(\mathbf{s} \mid F)} =$$

$$= \left[\frac{(F_a + 1)! F_b!}{(F_a + F_b + 2)!} \right] / \left[\frac{F_a! F_b!}{(F_a + F_b + 1)!} \right] =$$

$$= \frac{F_a + 1}{F_a + F_b + 2}$$

– выражение, известное как закон Лапласа.

Сравнение моделей

Предположим, что учёный предположил другую теорию для наших данных. Он настаивает, что источником является не погнутая монета, а идеально сформированный куб с шестью гранями, на одной из который указан орёл, а на других пяти – решки. Таким образом параметр p_a, который в исходной модели может принимать любое значение от 0 до 1, в новой гипотезе H_0 равен строго $1/6$.

Как можно сравнить эти две модели с использованием полученных данных. Мы хотим оценить насколько вероятна H_1 по сравнению с H_0.

Для сравнения моделей мы запишем теорему Байеса, однако в этот раз с другой переменной в левой части. Мы хотим знать, насколько вероятна гипотеза H_1 при имеющихся данных. По теореме Байеса

$$P(H_1 \mid \mathbf{s}, F) = \frac{P(\mathbf{s} \mid F, H_1) P(H_1)}{P(\mathbf{s} \mid F)}.$$

Аналогично, апостериорная вероятность H_0 равна

$$P(H_0 \mid \mathbf{s}, F) = \frac{P(\mathbf{s} \mid F, H_0) P(H_0)}{P(\mathbf{s} \mid F)}.$$

Нормализующая константа в обоих случаях $P(\mathbf{s} \mid F)$, которая представляет собой общую вероятность получения наблюдаемых данных. Если H_1 и H_0 единственные рассматриваемые модели, эта вероятность задаётся по правилу суммы

$$P(\mathbf{s} \mid F) = P(\mathbf{s} \mid F, H_1) P(H_1) + P(\mathbf{s} \mid F, H_0) P(H_0).$$

Чтобы вычислить апостериорные вероятности гипотез необходимо задать их априорные вероятности $P(H_1)$ и $P(H_0)$. В нашем случае, зададим их равными ½. Необходимо вычислить зависящие от данных значения $P(\mathbf{s} \mid F, H_1)$ и $P(\mathbf{s} \mid F, H_0)$. Мы можем назвать эти значения. Значение $P(\mathbf{s} \mid F, H_1)$ – это мера того, насколько

данные соответствуют H_1 и мы называем это значение свидетельством модели H_1. Это значение мы уже вычисляли ранее.

Свидетельство модели H_0 вычисляется намного проще, поскольку у модели нет оцениваемых параметров. Задавая p_0 равным $1/6$, получаем

$$P(\mathbf{s}\,|\,F,H_0) = p_0^{F_a}\left(1-p_0\right)^{F_b}.$$

Таким апостериорное отношение вероятностей модели H_1 к модели H_0 равно

$$\frac{P(H_1\,|\,\mathbf{s},F)}{P(H_0\,|\,\mathbf{s},F)} = \frac{P(\mathbf{s}\,|\,F,H_1)P(H_1)}{P(\mathbf{s}\,|\,F,H_0)P(H_0)} = \left[\frac{F_a!F_b!}{(F_a+F_b+1)!}\right] / \; p_0^{F_a}\left(1-p_0\right)^{F_b}.$$

Некоторые значения этого соотношения приведены в таблице 2. Обычно, для небольших размеров выборки вероятности моделей не слишком отличаются, однако чем больше данных, тем больше это отношение может быть.

Таблица 2. Апостериорное отношение вероятностей моделей

F_a	F_b	$P(a\,\|\,\mathbf{s},F)$	$\frac{P(H_1\,\|\,\mathbf{s},F)}{P(H_0\,\|\,\mathbf{s},F)}$	$\frac{P(H_0\,\|\,\mathbf{s},F)}{P(H_1\,\|\,\mathbf{s},F)}$
1	1	0,5	1,200	
2	1	0,6	3,600	
5	1	0,75	222,171	
10	1	0,846	549692,509	
1	2	0,4		1,389
1	5	0,25		2,813
1	20	0,087		2,008
4	4	0,5	4,266	
2	6	0,25		3,198
6	2	0,75	4886,156	

Тема 2. Энтропия[*]

Понятие энтропии

Первой целью нашего курса является определение некой количественной оценки информации, получаемой при наблюдении события, происходящего с заданной вероятностью.

Предположим, что задано дискретный вероятностный ансамбль $\{Z, p(z)\}$ с N возможными состояниями и заданным на нём распределением вероятностей $p(z_i)$ таким, что для всех $i = \overline{1,N}$ $p(z_i) \geq 0$, а $\sum p(z_i) = 1$:

$$Z = \begin{bmatrix} z_1, z_2, ..., z_i, ..., z_N \\ p_1, p_2, ..., p_i, ..., p_N \end{bmatrix}, \quad (1)$$

Первой мерой, введённой для определения количества информации, получаемой при наблюдении некоторой дискретной случайной величины была мера, предложенная Хартли:

$$I(Z) = \log_a N, \quad (2)$$

где N – число возможных исходов. b определяет единицу измерения информации, например, при b равном 2, единицей измерения информации будет бит, при 3 – трит, при e – нат, при 10 – дит, или хартли.

Отметим, что важным условием появления информации по Хартли является наличие нескольких возможных исходов. Очевидно, что если исход события гарантирован, мы не получаем информации от наблюдения.

В чём недостаток меры Хартли? Мера Хартли не учитывает того факта, что вероятности p_i, $i = \overline{1,N}$ в (1) могут быть различны. Поэтому она используется лишь в случае равновероятных событиях множества. При неравновероятных событиях неопределенность

[*] Темы 2, 3, 9, 15 и 16 содержат текст из учебного пособия «Теория информации» Фурсова В.А. и добавлены в настоящее учебное пособие с его согласия.

меньше. Например, неопределенность выбора в случае двух элементов с априорными вероятностями 0,9 и 0,1 меньше, чем в случае равновероятных элементов (0,5; 0,5). Поэтому естественным является требование, чтобы мера неопределенности была непрерывной функцией вероятностей p_i, $i=\overline{1,N}$ элементов. Удовлетворяющая этому требованию мера информации предложена К. Шенноном и называется энтропией:

$$H(Z)=-\sum_{i=1}^{N} p(z_i)\log_a p(z_i). \qquad (3)$$

Наиболее широко используется двоичная единица информации – бит, которая и будет использоваться далее.

Для независимо реализуемых элементов множества в качестве меры может использоваться априорная частная неопределенность:

$$H(z_i)=-\log_2 p(z_i). \qquad (4)$$

Нетрудно заметить, что мера К. Шеннона (3), характеризующая неопределённость источника в целом, получается усреднением частных неопределенностей (4) по всем элементам множества.

Покажем связь меры К. Шеннона с мерой Р. Хартли. Если все элементы множества равновероятны, т.е. $p_i=1/N$ для всех $i=\overline{1,N}$, то

$$H(Z)=-\sum_{i=1}^{N}\frac{1}{N}\log_2\frac{1}{N}=\log_2 N. \qquad (5)$$

Таким образом, мера Р. Хартли – частный случай меры К. Шеннона для равновероятных элементов. Можно также показать, что мера К. Шеннона является обобщением меры Хартли на случай неравновероятных элементов.

Свойства энтропии

1. Энтропия величина вещественная и неотрицательная. Свойство легко проверяется по формуле (3) с учетом того, что $0\le p(z_i)\le 1$ для всех $i=\overline{1,N}$.

2. Энтропия величина ограниченная. При $0 < p_i \le 1$ это свойство непосредственно следует из формулы (5). При $p = 0$ имеем:

$$\lim_{p \to 0} \left(-p \log_2 p\right) = \lim_{p \to 0} \frac{\log_2 \frac{1}{p}}{\frac{1}{p}} = \lim_{\alpha \to \infty} \frac{\log_2 \alpha}{\alpha} = \lim_{\alpha \to \infty} \frac{\log_2 e}{\alpha \cdot 1} = 0$$

(здесь произведена замена $1/p = \alpha$ и далее раскрыта неопределенность по правилу Лопиталя). Таким образом, при любых значениях $0 \le p_i \le 1$, $i = \overline{1, N}$ $H(Z) < \infty$.

3. По ходу доказательства свойства 2 нетрудно заметить, что $H(Z) = 0$, если вероятность одного из элементов множества равна 1.

4. Энтропия максимальна, когда все элементы множества равновероятны и

$$H_{\max}(Z) = \max_{\forall p_i} H(Z) = \log_2 N . \qquad (6)$$

Будем искать максимум (3) при условии $\sum p_i = 1$. Функция Лагранжа для соответствующей задачи на безусловный экстремум

$$F(p, \lambda) = -\sum_{p=1}^{N} p_i \log_2 p_i + \lambda \left(\sum_{i=1}^{N} p_i - 1 \right) \to extr .$$

Необходимые условия экстремума:

$$\frac{\partial F(p, \lambda)}{\partial p_i} = -\log_2 p_i - \log_2 e + \lambda = 0 ,$$

$$\frac{\partial F(p, \lambda)}{\partial \lambda} = \sum_{i=1}^{N} p_i - 1 = 0 ,$$

откуда следует $p_i = 2^{\lambda - \log_2 e} = Const = 1/N$. Проверкой легко убедиться, что указанное значение доставляет максимум.

5. В частном случае множества с двумя элементами зависимость энтропии от вероятности одного из элементов имеет вид, показанный на рисунке 2. В этом можно убедиться, применяя соотношения и

выводы, полученные при рассмотрении свойств 2 и 3 к соотношению (3), которое в данном случае принимает вид

$$H(Z) = -p\log_2 p - (1-p)\log_2 (1-p). \qquad (7)$$

В заключение подчеркнем, что энтропия характеризует только среднюю неопределенность выбора одного элемента из множества, полностью игнорируя их содержательную сторону.

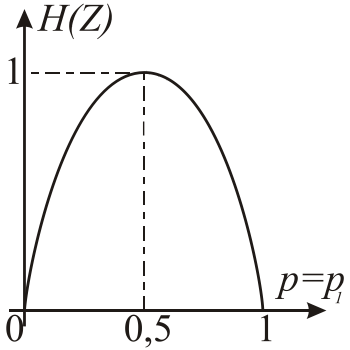

Рис. 2. Изменение энтропии в случае двух элементов

Понятие условной энтропии.

Рассмотрим теперь случай, когда заданы два множества: $Z = \{z_1, z_2, ..., z_N\}$ и $V = \{v_1, v_2, ..., v_K\}$, между элементами которых имеются связи. Произведением множеств $\{ZV\}$ называется множество, элементы которого представляют собой все возможные упорядоченные пары произведений $z_i v_j$, $i = \overline{1,N}$, $j = \overline{1,K}$. Если каждой паре z_i, v_j поставлена в соответствие вероятность $p(z_i, v_j)$, то имеем произведение ансамблей $\{ZV, p(zv)\}$. Для элементов объединенного ансамбля имеют место обычные свойства вероятностей:

$$\sum_{j=1}^{K} p(z_i, v_j) = p(z_i), \quad \sum_{i=1}^{N} p(z_i, v_j) = p(v_j). \qquad (8)$$

Из указанных свойств, в частности, следует, что если задано произведение ансамблей, то всегда могут быть найдены исходные

ансамбли $\{Z, p(z)\}$ и $\{V, p(v)\}$. Обратное возможно лишь в случае, когда элементы исходных ансамблей независимы, при этом $p(z_i, v_j) = p(z_i)\, p(v_j)$. Заметим, что поскольку при этом $\log_2 p(z_i, v_j) = \log_2 p(z_i) + \log_2 p(v_j)$ имеем

$$H(ZV) = -\sum_{i=1}^{N}\sum_{j=1}^{K} p(z_i, v_j)\log_2 p(z_i, v_j) =$$

$$= -\sum_{i=1}^{N}\sum_{j=1}^{K} p(z_i)\, p(v_j)\log_2\left[p(z_i)\, p(v_j)\right] =$$

$$= -\sum_{i=1}^{N} p(z_i)\log_2 p(z_i)\underbrace{\sum_{j=1}^{K} p(v_j)}_{1} - \sum_{j=1}^{K} p(v_j)\log_2 p(v_j)\underbrace{\sum_{i=1}^{N} p(z_i)}_{1} =$$

$$= H(Z) + H(V).$$

Аналогично могут быть получены формулы для объединения любого числа независимых источников.

В общем случае для зависимых ансамблей

$$p(z_i, v_j) = p(z_i)\, p(v_j / z_i) = p(v_j)\, p(z_i / v_j),$$

т.е. для определения вероятности элемента объединенного ансамбля необходимо задание условной вероятности появления элемента одного из ансамблей, при условии, что реализовался элемент другого ансамбля:

$$p(z_i / v_j) = \frac{p(z_i, v_j)}{p(v_j)}, \qquad p(v_j / z_i) = \frac{p(z_i, v_j)}{p(z_i)}.$$

Пусть объединенный ансамбль $\{ZV\}$ задан матрицей вероятностей всех его возможных элементов $z_i v_j$, $i = \overline{1, N}$, $j = \overline{1, K}$:

$$\begin{bmatrix} p(z_1, v_1) & p(z_2, v_1) & \dots & p(z_N, v_1) \\ p(z_1, v_2) & p(z_2, v_2) & \dots & p(z_N, v_2) \\ \dots & \dots & \dots & \dots \\ p(z_1, v_K) & p(z_2, v_K) & \dots & p(z_N, v_K) \end{bmatrix}. \qquad (9)$$

Суммируя вероятности по строкам и столбцам (9) в соответствии с (8) можно определить также ансамбли $\{Z, p(z)\}$ и $\{V, p(v)\}$:

$$\{Z, p(z)\} = \begin{bmatrix} z_1 & z_2 & \dots & z_N \\ p(z_1) & p(z_2) & \dots & p(z_N) \end{bmatrix},$$

$$\{V, p(v)\} = \begin{bmatrix} v_1 & v_2 & \dots & v_K \\ p(v_1) & p(v_2) & \dots & p(v_K) \end{bmatrix}.$$

Поскольку в случае зависимых элементов

$$p(z_i, v_j) = p(z_i)\, p(v_j / z_i) = p(v_i)\, p(z_i / v_j), \qquad (10)$$

с использованием первого из указанных в (1.10) равенств можно записать

$$\begin{aligned} H(ZV) &= -\sum_{ij} p(z_i, v_j) \log_2 p(z_i, v_j) = \\ &= -\sum_i p(z_i) \log_2 p(z_i) \sum_j p(v_j / z_i) - \\ &\quad -\sum_i p(z_i) \sum_j p(v_j / z_i) \log_2 p(v_j / z_i). \end{aligned} \qquad (11)$$

По условию нормировки $\sum_j p(v_j / z_i) = 1$ для любого $i = \overline{1, N}$, поэтому первое слагаемое в правой части является энтропией $H(Z)$ ансамбля $\{Z, p(z)\}$. Вторая сумма (по j) во втором слагаемом характеризует частную неопределенность, приходящуюся на одно состояние ансамбля V при условии, что реализовалось состояние z_i ансамбля Z. Ее называют частной условной энтропией и обозначают $H_{z_i}(V)$:

$$H_{z_i}(V) = -\sum_{j=1}^{K} p(v_j / z_i) \log_2 p(v_j / z_i). \qquad (12)$$

Величина $H_Z(V)$, получаемая усреднением частной условной энтропии по всем элементам z_i:

$$H_Z(V) = \sum_{i=1}^{N} p(z_i) H_{z_i}(V), \qquad (13)$$

называется полной условной энтропией или просто условной энтропией. Таким образом, (11) с учетом (12), (13) можно записать в виде

$$H(ZV) = H(Z) + H_Z(V). \qquad (14)$$

Используя второе равенство в (4.12), по аналогии можно записать:

$$H(ZV) = H(V) + H_V(Z). \qquad (15)$$

Можно также показать, что в случае объединения любого числа множеств $\{ZVW...\}$ с зависимыми элементами имеет место равенство

$$H(ZVW...) = H(Z) + H_Z(V) + H_{ZV}(W) + ... \quad .$$

Подчеркнем, что условная энтропия всегда меньше или равна безусловной:

$$H_V(Z) \le H(Z), \;\; H_Z(V) \le H(V). \qquad (16)$$

Справедливость неравенств (16) интуитивно понятна: неопределенность выбора элемента из некоторого множества может только уменьшиться, если известен элемент другого множества, с элементами которого существует взаимосвязь. Из (14)–(16), в частности, следует

$$H(ZV) \le H(Z) + H(V). \qquad (17)$$

Часто имеет место другой тип связи, а именно: статистическая зависимость между элементами последовательности. Если имеет место связь только между двумя соседними элементами последовательности, она характеризуется условной вероятностью $p(z_i / z_j)$. Последовательность элементов, обладающую указанным свойством, называют односвязной цепью Маркова. Связь каждого элемента с двумя предшествующими характеризуется условной вероятностью $p(z_i / z_j z_k)$, а соответствующая последовательность называется двусвязной цепью Маркова.

Для односвязной цепи Маркова в предположении, что известен (принят) элемент z_j из алфавита объема N, частная условная энтропия

$$H\left(Z / z_j\right) = -\sum_{i=1}^{N} p\left(z_i / z_j\right) \log_2 p\left(z_i / z_j\right).$$

При этом полная (средняя) условная энтропия определяется как

$$H(Z) = -\sum_{j=1}^{N} p\left(z_j\right) \sum_{i=1}^{N} p\left(z_i / z_j\right) \log_2 p\left(z_i / z_j\right). \qquad (18)$$

Аналогично для двусвязной цепи Маркова

$$H\left(Z / z_j z_k\right) = -\sum_{i=1}^{N} p(z_i / z_j z_k) \log_2 p(z_i / z_j z_k),$$

$$H(Z) = -\sum_{j,k} p(z_j, z_k) \sum_{i} p(z_i / z_j z_k) \log_2 p(z_i / z_j z_k). \qquad (19)$$

Можно построить выражения для энтропии и при более протяженной связи между элементами последовательности.

Понятие дифференциальной энтропии

Перейдем к рассмотрению источников информации, выходные сигналы которых являются непрерывной случайной величиной. Множество возможных состояний такого источника составляет континуум, а вероятность любого конкретного значения равна 0, что делает невозможным применение, например, меры (3). Построим меры неопределенности таких источников, опираясь на введенные ранее меры для дискретных ансамблей.

Мы можем приближенно оценить неопределенность выбора какого-либо значения непрерывной случайной величины по формуле (3), если ограничим диапазон ее допустимых значений и разобьем этот диапазон, например, на равные интервалы, вероятность попадания в каждый из которых отлична от нуля и определяется как

$$\mathrm{P}\left\{z_i \le Z < z_i + \Delta z\right\} \cong w\left(z_i^*\right) \Delta z.$$

Здесь $w\left(z_i^*\right)$ – ордината плотности распределения $w(z)$ непрерывной случайной величины при значении z_i^*, принадлежащем интервалу $\left[z_i, z_i + \Delta z\right]$.

Заменяя в (3) $w(z_i)$ его приближенным значением $w\left(z_i^*\right) \cdot \Delta z$ имеем

$$H(Z) = -\sum_{i=1}^{N} w\left(z_i^*\right)\Delta z \log_2 \left(w\left(z_i^*\right)\Delta z\right) = \\ = -\sum_{i=1}^{N} w\left(z_i^*\right)\log_2 w\left(z_i^*\right)\Delta z - \log_2 \Delta z \sum_{i=1}^{N} w\left(z_i^*\right)\Delta z. \qquad (20)$$

Далее осуществим предельный переход при $\Delta z \to 0$. При этом сумма переходит в интеграл, $\Delta z \to dz$, а $\sum_{i=1}^{N} w(z_i^*)\Delta z \to 1$. С учетом того, что в общем случае диапазон изменения непрерывной случайной величины $(-\infty; +\infty)$, получаем:

$$H(Z) = -\int_{-\infty}^{+\infty} w(z)\log_2 w(z)\,dz - \lim_{\Delta z \to 0} \log_2 \Delta z\,. \qquad (21)$$

Из формулы (21) следует, что энтропия непрерывной случайной величины равна бесконечности независимо от вида плотности вероятности. Этот факт, вообще говоря, не является удивительным, так как вероятность конкретного значения непрерывного сигнала равна 0, а множество состояний бесконечно. Ясно, что использовать такую меру на практике не представляется возможным.

Для получения конечной характеристики информационных свойств используется только первое слагаемое, называемое дифференциальной энтропией:

$$h(Z) = -\int_{-\infty}^{+\infty} w(z)\log_2 w(z)\,dz\,. \qquad (22)$$

Термин дифференциальная энтропия связан с тем, что для ее определения в формуле (22) используется дифференциальный закон

распределения $p(z)$. Возникает естественный вопрос: не является ли это соглашение искусственным и не имеющим смысла.

Оказывается, что дифференциальная энтропия имеет смысл средней неопределённости выбора случайной величины с произвольным законом распределения за вычетом неопределённости случайной величины, равномерно распределённой в единичном интервале.

Действительно энтропия (2.2) равномерно распределённой на интервале δ случайной величины Z_r определяется как

$$H(Z_r) = -\int_{-\infty}^{\infty} \frac{1}{\delta} \log_2 \frac{1}{\delta} dz - \lim_{\Delta z \to 0} \log_2 \Delta z_r .$$

При $\delta = 1$:

$$H(Z_r) = -\lim_{\Delta z \to 0} \log_2 \Delta z_r . \qquad (23)$$

Сравнивая (22) и (23) нетрудно заметить, что при $\Delta z = \Delta z_r$

$$H(Z) - H(Z_r) = h(z). \qquad (24)$$

Понятие дифференциальной условной энтропии

Рассмотрим теперь ситуацию, когда (далее две) непрерывные случайные величины статистически связаны. Как и ранее разобьем диапазоны допустимых значений случайных величин на равные интервалы так, что

$$\mathrm{P}\{z_i \le Z < z_i + \Delta z, \quad v_j \le V < v_j + \Delta v\} \cong w(z_i^*, v_j^*) \cdot \Delta z \Delta v , \qquad (25)$$

где $w(z_i^*, v_j^*)$ – ордината двумерной плотности распределения в точке (z_i^*, v_j^*), принадлежащей прямоугольнику со сторонами $\Delta z, \quad \Delta v$: $(z_i \le z_i^* < z_i + \Delta z, \qquad v_j \le v_j^* < v_j + \Delta v)$. Подставляя приближенные значения вероятностей (2.6) в формулу для энтропии (1.3) получаем

$$H(Z,V) = -\sum_i \sum_j w(z_i^*, v_j^*) \log_2 w(z_i^*, v_j^*) \Delta z \Delta v -$$

$$- \log_2 \Delta z \sum_i \sum_j w(z_i^*, v_j^*) \Delta z \Delta v - \log_2 \Delta v \sum_i \sum_j w(z_i^*, v_j^*) \Delta z \Delta v.$$

С учетом того, что $w(z_i^*, v_j^*) = w(z_i^*) w(v_j^* / z_i^*)$ первое слагаемое в правой части последнего равенства можно представить в виде суммы $-\sum_i w(z_i^*) \log_2 w(z_i^*) \Delta z \sum_j w(v_j^* / z_i^*) \Delta v - \sum_i \sum_j w(z_i^*, v_j^*) \log_2 w(v_j^* / z_i^*) \Delta v \Delta z$.

Далее осуществляя предельный переход при $\Delta z \to 0, \quad \Delta v \to 0$, с учетом того, что по условию нормировки

$$\lim_{\substack{\Delta z \to 0 \\ \Delta v \to 0}} \sum_i \sum_j w(z_i^*, v_j^*) \Delta z \Delta v = 1,$$

$$\lim_{\Delta v \to 0} \sum_i \sum_j w(v_j^* / z_i^*) \Delta v = 1,$$

$$\lim_{\Delta z \to 0} \sum_i \sum_j w(z_i^*) \Delta z = 1,$$

получаем

$$H(Z,V) = -\int_{-\infty}^{\infty} w(z) \log_2 w(z) dz - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} w(z,v) \log_2 w(v/z) dz dv - \quad (26)$$

$$- \lim_{\Delta z \to 0} \log_2 \Delta z \quad - \lim_{\Delta v \to 0} \log_2 \Delta v.$$

Первое и третье слагаемое – суть энтропия $H(Z)$ непрерывного источника (22), выходным сигналом которого является случайная величина Z, а величина

$$H_Z(V) = -\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} w(z,v) \log_2 w(v/z) dz dv - \lim_{\Delta v \to 0} \log_2 \Delta v$$

является условной энтропией непрерывной случайной величины. Она, как и следовало ожидать, в силу второго слагаемого в правой части равна бесконечности. Поэтому, как и в случае одного независимого источника, принимают во внимание только первое слагаемое:

$$h_Z(V) = -\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} w(z,v)\log_2\frac{w(z,v)}{w(z)}dzdv\,. \qquad (27)$$

Величину (27) называют условной дифференциальной энтропией.

Условная дифференциальная энтропия характеризует среднюю неопределенность выбора непрерывной случайной величины с произвольным законом распределения при условии, что известны результаты реализации другой, статистически связанной с ней непрерывной случайной величины, за вычетом средней неопределенности выбора случайной величины, имеющей равномерное распределение на единичном интервале.

Дифференциальную энтропию двух непрерывных статистически связанных источников можно представить в виде

$$h(ZV) = h(Z) + h_Z(V) = h(V) + h_V(Z)\,. \qquad (28)$$

Второе равенство в (2.10) получается по той же схеме, что и первое, при $w(z_i^*, v_j^*) = w(v_j^*)w(z_i^* / v_j^*)$. Заметим также, что в соответствии с (2.7), (2.8) для непрерывных источников можно выписать равенства, аналогичные (1.14) и (1.16) для дискретных сообщений: $H(ZV) = H(Z) + H_Z(V) = H(V) + H_V(Z)$, однако они имеют лишь теоретическое значение, поскольку оперировать на практике с бесконечными неопределенностями не представляется возможным.

Свойства дифференциальной энтропии

Дифференциальная энтропия в отличие от энтропии дискретного источника является относительной мерой неопределенности, т.к. её значения зависят от масштаба непрерывной величины. Действительно, предположим, что непрерывная случайная величина Z изменилась в k раз. Поскольку всегда должно выполняться условие нормировки:

$$\int_{-\infty}^{+\infty} w(kz)d(kz) = k\int_{-\infty}^{+\infty} w(kz)dz = 1,$$

имеет место следующее соотношение для плотностей исходной и масштабированной величин

$$w(kz) = \frac{w(z)}{k}. \qquad (29)$$

С учетом (29) в соответствии с (22) имеем

$$h(kZ) = -\int_{-\infty}^{+\infty} w(kz) \cdot \log_2 w(kz) \cdot d(kz) =$$

$$= -\int_{-\infty}^{+\infty} w(z)\left[\log_2 w(z) - \log_2 k\right] dz = \qquad (30)$$

$$= -\int_{-\infty}^{+\infty} w(z) \log_2 w(z) dz + \log_2 k \int_{-\infty}^{+\infty} w(z) dz = h(Z) + \log_2 k.$$

Из (30) следует, что из-за выбора различных k дифференциальная энтропия может принимать положительные, отрицательные и нулевые значения.

Дифференциальная энтропия не зависит от параметра сдвига $\Theta = Const$, т.е. $h(Z + \Theta) = h(Z)$. Действительно, используя замену $V = Z + \Theta$, при которой пределы интегрирования не изменяются, а $dz = dv$ имеем:

$$h(Z + \Theta) = -\int_{-\infty}^{+\infty} w(z + \Theta) \log_2 w(z + \Theta) dz =$$

$$= -\int_{-\infty}^{+\infty} w(v) \log_2 w(v) dv = h(V).$$

Распределения, обладающие максимальной дифференциальной энтропией

Сформулируем следующую задачу. Определить плотность $p(z)$, обеспечивающую максимальное значение функционала

$$h(Z) = -\int_{\alpha}^{\beta} w(z) \log_2 w(z) dz,$$

при ограничении

$$\int_{\alpha}^{\beta} w(z)dz = 1.$$

Функция Лагранжа в указанной (изопериметрической) задаче имеет вид

$$F(w,\mu) = w(z)\log_2 w(z) + \mu \cdot w(z), \qquad (31)$$

где μ, в данном случае постоянный, неопределенный множитель Лагранжа. Необходимые условия экстремума (31) даются соотношением

$$\frac{\partial F(w,\mu)}{\partial w} = \log_2 w(z) + \log_2 e + \mu = 0. \qquad (32)$$

Искомая плотность $w(z) = 1/(\beta - \alpha)$, $\alpha \le z \le \beta$ получается в результате совместного решения (31), (32). Это означает, что если единственным ограничением для случайной величины является область возможных значений: $Z \in [\alpha, \beta]$, то максимальной дифференциальной энтропией обладает равномерное распределение вероятностей в этой области.

Снимем теперь ограничение на область возможных значений, но добавим ограничение на величину дисперсии:

$$h(Z) = -\int_{-\infty}^{\infty} w(z)\log_2 w(z)dz \to max, \qquad (33)$$

при

$$\int_{-\infty}^{\infty} w(z)dz = 1, \qquad (34)$$

$$\int_{-\infty}^{\infty} z^2 w(z)dz = \sigma^2. \qquad (35)$$

Функция Лагранжа в данном случае принимает вид

$$F(w,\mu_1,\mu_2) = w(z)\log_2 w(z) + \mu_1 \cdot w(z) + \mu_2 z^2 w(z),$$

а соответствующее уравнение Эйлера

$$\frac{\partial F(w,\mu)}{\partial p} = \log_2 w(z) + \log_2 e + \mu_1 + \mu_2 z^2 = 0. \qquad (36)$$

Непосредственной подстановкой можно убедиться, что гауссовская плотность

$$w(z) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{z^2}{2\sigma^2} \right\}$$

удовлетворяет необходимому условию (36) экстремума (в данном случае максимума) функционала (33) и заданным изопериметрическим ограничениям (34), (35). Заметим, что при выводе для простоты математическое ожидание мы приняли равным нулю, поскольку дифференциальная энтропия все равно не зависит от параметра сдвига.

Тема 3. Количество информации

Количество информации при передаче отдельного элемента дискретного сообщения

Предположим, что задан некоторый дискретный источник информации, характеризующийся дискретным вероятностным ансамблем:

$$Z = \begin{bmatrix} z_1 & z_2 & \cdots & z_N \\ p(z_1) & p(z_2) & \cdots & p(z_N) \end{bmatrix},$$

где z_i, $i = \overline{1, N}$ – его возможные состояния. Каждому состоянию источника можно поставить в соответствие отдельный первичный сигнал. Некоторую заданную совокупность первичных сигналов, поступающих с выхода источника информации на вход канала связи принято называть сообщением, а z_i – элементом сообщения.

Если состояния источника реализуются независимо друг от друга, то частная априорная неопределённость появления на входе канала элемента сообщения z_i определяется как

$$H(z_i) = -\log_2 p(z_i).$$

Предположим, что статистическая связь между помехой и элементами сообщения отсутствует и известна условная вероятность того, что вместо z_i принимается v_j:

$$p(z_i / v_j), \quad i = \overline{1, N}, \quad j = \overline{1, K}.$$

Таким образом, если на выходе канала получен элемент v_j, то становится известной апостериорная вероятность $p(z_i / v_j)$. Следовательно, можно определить апостериорную частную неопределённость:

$$H_{v_j}(z_i) = -\log_2 p(z_i / v_j).$$

Частное количество информации, полученное в результате того, что стал известен элемент v_j, определим как разность априорной и апостериорной неопределенностей:

$$I\left(z_i, v_j\right) = H\left(z_i\right) - H_{v_j}\left(z_i\right) =$$

$$= -\log_2 p\left(z_i\right) + \log_2 p\left(z_i / v_j\right) = \log_2 \frac{p\left(z_i / v_j\right)}{p\left(z_i\right)}. \qquad (37)$$

Таким образом, частное количество информации равно величине неопределённости, которая снята в результате получения элемента сообщения v_j.

Свойства частного количества информации

1. Частное количество информации уменьшается с ростом априорной вероятности $p\left(z_i\right)$, увеличивается с ростом апостериорной вероятности $p\left(z_i / v_j\right)$ и в зависимости от соотношения между ними может быть положительным, отрицательным и нулевым (свойство непосредственно следует из (3.3).

2. Если $p\left(z_i / v_j\right) = p\left(z_i\right)$, то в соответствии с (37) $I\left(z_i, v_j\right) = 0$.

3. При отсутствии помехи частное количество информации равно частной априорной неопределенности элемента z_i: $I(z_i, v_j) = H(z_i) = -\log_2 p(z_i)$, поскольку при этом $H_{v_j}\left(z_i\right) = 0$.

4. Частное количество информации о z_i, содержащееся в v_j, равно частному количеству информации о v_j, содержащемуся в z_i. Действительно:

$$I(z_i, v_j) = \log_2 \frac{p\left(z_i / v_j\right)}{p(z_i)} = \log_2 \frac{p(v_j)\, p\left(z_i / v_j\right)}{p(v_j)\, p(z_i)} =$$

$$\log_2 \frac{p(z_i)\, p\left(v_j / z_i\right)}{p(z_i)\, p(v_j)} = \log_2 \frac{p\left(v_j / z_i\right)}{p(v_j)} = I(v_j, z_i).$$

Среднее количество информации в любом элементе дискретного сообщения

Априорная неопределённость в среднем на один элемент сообщения характеризуется энтропией:

$$H(Z) = -\sum_{i=1}^{N} p(z_i) \cdot \log_2 p(z_i), \qquad (38)$$

а апостериорная неопределенность – условной энтропией:

$$H_V(Z) = -\sum_{j=1}^{K} p(v_j) \sum_{i=1}^{N} p\left(z_i / v_j\right) \log_2 p\left(z_i / v_j\right). \qquad (39)$$

В соответствии с (38), (39) по аналогии с частным количеством информации количество информации в среднем на один элемент сообщения определим как

$$I(Z,V) = H(Z) - H_V(Z) =$$

$$= -\sum_i p(z_i) \log_2 p(z_i) + \sum_j p(v_j) \sum_i p\left(z_i / v_j\right) \log_2 p\left(z_i / v_j\right).$$

В последнем равенстве ничего не изменится, если первое слагаемое в правой части умножить на $\sum_{j=1}^{K} p\left(v_j / z_i\right) = 1$. Тогда, с учетом того, что

$$\sum_i p(z_i) \sum_j p\left(v_j / z_i\right) = \sum_j p(v_j) \sum_i p\left(z_i / v_j\right) = \sum_{ij} p\left(z_i, v_j\right)$$

и используя свойства логарифма, формулу для количества информации в среднем на один элемент сообщения можно записать в виде

$$I(Z,V) = \sum_{ij} p(z_i, v_j) \log_2 \frac{p\left(z_i / v_j\right)}{p(z_i)} = \sum_{ij} p(z_i, v_j) \log_2 \frac{p\left(z_i, v_j\right)}{p(z_i)\, p(v_j)} \qquad (40)$$

Далее, если частный характер количества информации не будет оговариваться специально, то всегда будет подразумеваться количество информации в среднем на один элемент сообщения (40).

Свойства среднего количества информации в элементе сообщения

1. Неотрицательность. $I(Z,V) \geq 0$, так как всегда $H(Z) \geq H_V(Z)$.

2. $I(Z,V) = 0$ при отсутствии статистической связи между Z и V, так как при этом $H(Z) = H_V(Z)$.

3. $I(Z,V) = I(V,Z)$, то есть количество информации в V относительно Z равно количеству информации в Z относительно V. Действительно

$$I(Z,V) - I(V,Z) = H(Z) - H_V(Z) - (H(V) - H_Z(V)) =$$
$$= H(Z) + H_Z(V) - (H(V) + H_V(Z)) = H(Z,V) - H(V,Z) = 0.$$

4. При отсутствии помех $I(Z,V) = H(Z)$, поскольку при этом $H_V(Z) = 0$. Это максимальное количество информации, которое может быть получено от источника.

Количество информации при передаче сообщений от непрерывного источника

Соотношение для количества информации от непрерывного источника получим из формулы (40) для дискретного случая. Обозначив переданный и принятый непрерывные сигналы соответственно Z и V разобьем область допустимых значений этих сигналов на равные интервалы и запишем приближенные вероятности (см. рисунок 3):

$$\mathrm{P}\left\{z_i \leq Z < z_i + \Delta z, v_j \leq V < v_j + \Delta v\right\} \cong w\left(z_i^*, v_j^*\right)\Delta z \Delta v,$$

где $w\left(z_i^*, v_j^*\right)$ – ордината двумерной плотности распределения $w(z,v)$ в некоторой точке, принадлежащей прямоугольнику с номером i, j.

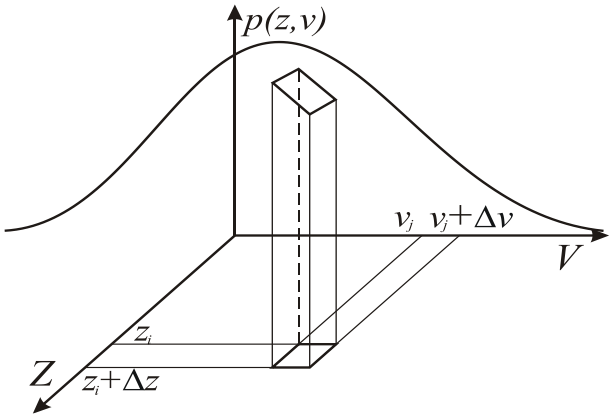

Рис. 3. Дискретизация области Z,V

Для соответствующих заданной двумерной плотности $w(z,v)$ одномерных плотностей $w(z_i)$, $w(v_j)$, по аналогии с тем как мы поступали при получении соотношения для дифференциальной энтропии, можно записать

$$\mathrm{P}\{z_i \le Z < z_i + \Delta z\} \cong w(z_i^*)\Delta z,$$

$$\mathrm{P}\{v_j \le V < v_j + \Delta v\} \cong w(v_j^*)\Delta v,$$

где $w(z_i^*)$, $w(v_j^*)$ – ординаты одномерных плотностей для значений z_i^* и v_j^*, взятых в интервалах $[z_i, z_i + \Delta z]$ и $[v_i, v_i + \Delta v]$ соответственно.

Заменяя в (3.6) $w(z_i, v_j)$, $w(z_i)$, $w(v_j)$ их приближенными значениями $w(z_i^*, v_j^*)\Delta z\Delta v$, $w(z_i^*)\Delta z$, $w(v_j^*)\Delta v$ соответственно, можно записать

$$I(Z,V) = \sum_i \sum_j w(z_i^*, v_j^*)\Delta z\Delta v \cdot \log_2 \frac{w(z_i^*, v_j^*)}{w(z_i^*)\,p(v_j^*)}. \qquad (41)$$

Осуществляя в (41) предельный переход при $\Delta z \to 0$, $\Delta v \to 0$ получаем:

$$I(Z,V)=\lim_{\substack{\Delta z\to 0\\ \Delta v\to 0}}\sum_i\sum_j w^*(z_i,v_j)\log_2\frac{w^*(z_i,v_j)}{w^*(z_i)p^*(v_j)}\Delta z\Delta v=$$
$$=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}w(z,v)\log_2\frac{w(z,v)}{w(z)w(v)}dzdv. \qquad (42)$$

Формула (42) может быть получена также с использования понятия дифференциальной энтропии. Действительно по аналогии с дискретным случаем определим количество информации как разность априорной и апостериорной (в данном случае дифференциальной) энтропии:

$$I(Z,V)=h(Z)-h_v(Z)=$$
$$=-\int_{-\infty}^{\infty}w(z)\log_2 w(z)dz+\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}w(z,v)\log_2 w\left(z/v\right)dzdv. \qquad (43)$$

В (43) ничего не изменится, если первое слагаемое в правой части умножить на $\int_{-\infty}^{\infty}w(v/z)dv=1$. Тогда, с учетом того, что $w(z,v)=w(v)w(z/v)=\quad=w(z)w(v/z)$, соотношение (43) можно переписать в следующем виде:

$$I(Z,V)=h(Z)-h_v(Z)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}w(z,v)\log_2\frac{w(z,v)}{w(z)w(v)}dzdv\,. \qquad (44)$$

Поскольку $I(Z,V)$ в (44) определяется как разность $h(Z)-h_v(Z)$, количество информации при передаче от непрерывного источника, в отличие от дифференциальной энтропии, уже не зависит от масштаба случайной величины. Заметим, что соотношение между понятиями энтропии и количества информации для непрерывного источника информации подобно соотношению между потенциалом, определяемым как работа по перенесению заряда из бесконечности в данную точку поля, и напряжением, определяемым как разность потенциалов, которое рассматривается в физике.

Тема 4. Каналы передачи данных.

Дискретный канал без памяти.

Дискретный канал без памяти характеризуется следующими величинами:

- входной алфавит A,
- выходной алфавит B,
- условные вероятности $P_{Y|X}(\cdot \mid x)$ для всех $x \in A$

$$P(y_n \mid x_1,...,x_{n-1}, y_1,...,y_{n-1}) = P_{Y|X}(y_n \mid x_n).$$

Из представленного выражения можно видеть, что вероятность появления значения y_n на выходе для заданного входа x_n и предыдущих значений выхода $y_1,...,y_{n-1}$ зависит только от текущего значения входа x_n. Такой канал называется каналом без памяти, поскольку значения выхода не зависят от предыдущих значений выхода и выхода, только от текущего входного значения.

Дискретный канал без памяти и без обратной связи описывается следующей вероятностью

$$P(x_n \mid x_1,...,x_{n-1}, y_1,...,y_{n-1}) = P(x_n \mid x_1,...,x_{n-1}).$$

Поскольку обратной связи нет, входные значения x_n не зависят от предыдущих выходных значений $x_1,...,x_{n-1}$.

Теорема.

Для дискретного канала без памяти без обратной связи

$$P(y_1,...,y_n \mid x_1,...,x_n) = \prod_{i=1}^{n} P_{Y|X}(y_i \mid x_i), \quad n = 1,2,... .$$

Доказательство:

Зададим совместную вероятность $P(x_1,...,x_n, y_1,...,y_n)$. Эта вероятность может быть записана как

$$P(x_1,...,x_n,y_1,...,y_n)=P(x_1,...,x_n,y_1,...,y_{n-1})\cdot P(y_n\mid x_1,...,x_n,y_1,...,y_{n-1})=$$
$$=P(x_1,...,x_{n-1},y_1,...,y_{n-1})\cdot P(x_n\mid x_1,...,x_{n-1},y_1,...,y_{n-1})\cdot$$
$$\cdot P(y_n\mid x_1,...,x_n,y_1,...,y_{n-1})=$$
$$=\prod_{i=1}^{n}P(x_i\mid x_1,...,x_{i-1},y_1,...,y_{n-1})\cdot P(y_i\mid x_1,...,x_i,y_1,...,y_{i-1}).$$

Из определения дискретного канала без памяти и без обратной связи, множители под знаком произведения могут быть записаны как

$$P(x_1,...,x_n,y_1,...,y_n)=\prod_{i=1}^{n}P(x_i\mid x_1,...,x_{i-1})\cdot P_{Y|X}(y_i\mid x_1).$$

Далее это выражение можно упростить, разбив произведение на два

$$P(x_1,...,x_n,y_1,...,y_n)=\left[\prod_{i=1}^{n}P(x_i\mid x_1,...,x_{i-1})\right]\cdot\left[\prod_{i=1}^{n}P_{Y|X}(y_i\mid x_1)\right].$$

Первая часть этого выражения равна $P(x_1,...,x_n)$

$$P(x_1,...,x_n,y_1,...,y_n)=P(x_1,...,x_n)\cdot\left[\prod_{i=1}^{n}P_{Y|X}(y_i\mid x_1)\right].$$

Разделив обе части выражения на $P(x_1,...,x_n)$ и воспользовавшись определением условной вероятности, получим:

$$\frac{P(x_1,...,x_n,y_1,...,y_n)}{P(x_1,...,x_n)}=\prod_{i=1}^{n}P_{Y|X}(y_i\mid x_1),$$

$$P(y_1,...,y_n\mid x_1,...,x_n)=\prod_{i=1}^{n}P_{Y|X}(y_i\mid x_1).$$

Пропускная способность канала без памяти определяется как максимум средней взаимной информации $I(X,Y)$, которая может быть получена посредством выбора $P(x)$, то есть

$$C=\max_{P_x}I(X,Y).$$

Это максимум взаимной информации между входом и выходом канала, при том, что максимизация производится по всем возможным распределения входа P_x. Полностью эквивалентно, из определения количества информации, пропускная способность может быть записана как

$$C = \max_{P_x} \left[H(X) - H(Y \mid X) \right].$$

Равномерно диспергирующий канал

Пусть у дискретного канала без памяти есть K возможных значений на входе и J на выходе. Назовём канал равномерно диспергирующим, если переходные вероятности, упорядоченные в порядке убывания одни и те же для каждого из K входов.

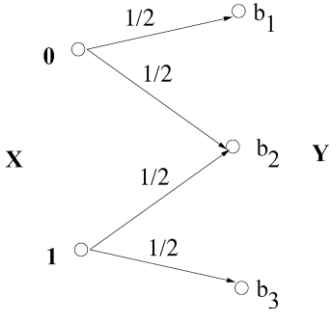

Рис. 4. Равномерно дипергирующий канал

Теорема.

Независимо от выбора распределения P_x для РДК

$$H(Y \mid X) = -\sum_{j=1}^{J} p_j \log p_j .$$

где $p_1, p_2, \ldots, p_j$ переходные вероятности.

Доказательство:

Из определения РДК известно, что условная энтропия выходного символа Y для заданного входа $X = a_k$ равна

$$H(Y \mid X = a_k) = -\sum_{j=1}^{J} p_j \log p_j, \quad k = 1, \ldots, K$$

Из определения $H(Y \mid X)$ следует

$$H(Y \mid X) = \sum_{k=1}^{K} P(a_k) H(Y \mid X = a_k).$$

Поскольку $H(Y \mid X = a_k)$ постоянна вне зависимости от a_k из-за РДК, выражение принимает вид

$$H(Y \mid X) = H(Y \mid X = a_k) \sum_{k=1}^{K} P(a_k) = H(Y \mid X = a_k).$$

Поэтому

$$H(Y \mid X) = -\sum_{j=1}^{J} p_j \log p_j .$$

Таким образом

$$C = \max_{P_x} H(Y) + \sum_{j=1}^{J} p_j \log p_j .$$

Равномерно фокусирующий канал

Рассмотрим дискретный канал без памяти с K входных значений и J выходных. Назовём этот канал равномерно фокусирующим, если переходные вероятности, упорядоченные в порядке убывания одни и те же для каждого из J выходов.

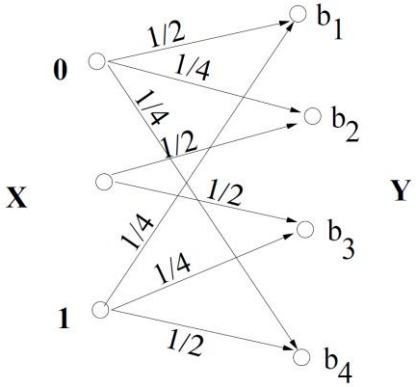

Рис. 5. Равномерно фокусирующий канал

Теорема.

В равномерно фокусирующем канале равномерные вероятности входных значений приводят к равномерным распределениям вероятностей выходов. При этом

$$\max_{P_x} H(Y) = \log J .$$

Доказательство:

Поскольку входное распределение $P(x)$ равномерно распределённое с K возможными значениями, можно записать следующее равенство

$$P(y) = \sum_x P(y \mid x) P(x) = \frac{1}{K} \sum_x P(y \mid x) .$$

Слагаемые $P(y \mid x)$ соответствуют вероятностям перехода в каждое значение y из всех возможных входов. Из определения равномерно фокусирующего канала, $P(y \mid x)$ постоянны для всех значений y. Следовательно, сумма в правой части будет одной и той же для всех J значений y, а значит, Y имеет равномерное распределение. Поскольку его распределение равномерно, исходя из свойства энтропии, получаем:

$$\max_{P_x} H(Y) = \log J .$$

Сильно-симметричный канал

Дискретный канал без памяти, который одновременно равномерно диспергирующий и равномерно фокусирующий называется сильно-симметричным каналом.

Бинарный симметричный канал – симметричный канала, в котором число входов и выходов равно двум.

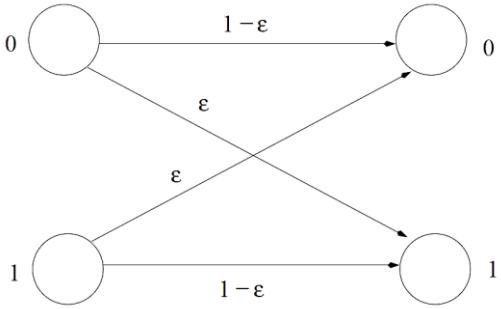

Рис. 6. Бинарный сильно-симметричный канал

Теорема.

Пропускная способность сильно-симметричного канала равна

$$C = \max_{P_x} H(Y) + \sum_{j=1}^{J} p_j \log p_j \ .$$

Из определения равномерно фокусирующего канала

$$\max_{P_x} H(Y) = \log J \ .$$

Комбинируя эти выражения, получаем пропускную способность сильно-симметричного канала

$$C = \log J + \sum_{j=1}^{J} p_j \log p_j \ .$$

Для бинарного симметричного канала $J = 2$ и его пропускная способность равна

$$C = 1 - h(\varepsilon) \ .$$

Симметричный канал

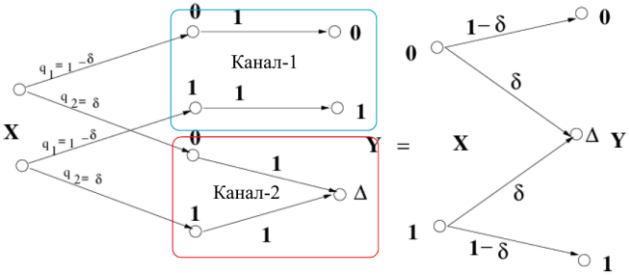

Рис. 7. Симметричный канал

Существует класс каналов, которые могут быть декомпозированы на L сильно-симметричных каналов с выбором вероятностей $q_1, q_2, ..., q_L$.

Алгоритм декомпозиции симметричного канала на сильно-симметричные.

1. Разбить набор выходных символов на подмножества $B^{(1)}, B^{(2)}, ..., B^{(L)}$ таким образом, чтобы любые два выходных символа находились в одном подмножестве тогда и только тогда, если они «одинаково сфокусированы».
2. Проверить, являются ли все выходных символы «одинаково диспергированными» относительно каждого символа в подмножестве $B^{(i)}$. Если да, присвоить этому подмножеству вероятность q_i, равную сумме вероятностей попадания в $B^{(i)}$. Если нет, канал не симметричный.
3. Если определены все q_i, канал симметричный, декомпозиция произведена.

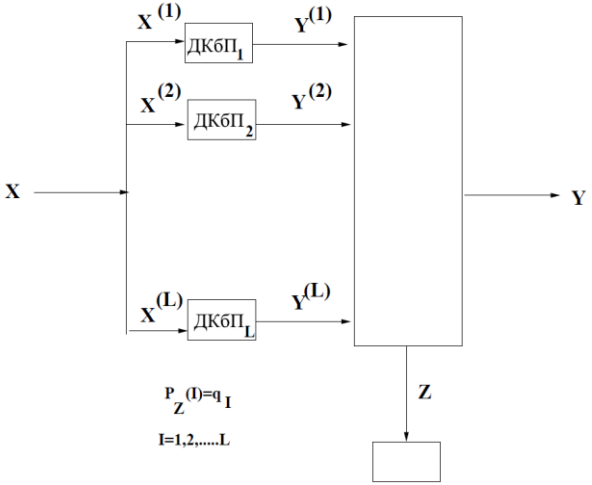

Рис. 8. Декомпозиция симметричного канала

Схема демонстрирует общую блочную диаграмму декомпозиции симметричного дискретного канала без памяти. У нас есть вход X и

L сильно-симметричных каналов, выбираемых с вероятностями $q_1, q_2, ..., q_L$, при этом выбор контролируется Z.

Сделаем следующие предположения:

- Все составные каналы имеют один и тот же входной алфавит, то есть входные значения каждого канала выбираются из одних и тех же символов;
- Все составные каналы имеют различные непересекающиеся выходные алфавиты;
- X и Z статистически независимы, то есть значение входного символа не влияет на выбор составного канала;
- Вероятности выбора канала равны $q_1, q_2, ..., q_L$.

Теорема: Для симметричного дискретного канала без памяти, удовлетворяющего перечисленным требованиям

$$I(X,Y) = \sum_{i=1}^{L} I\left(X, Y^{(i)}\right) q_i \ .$$

Доказательство:

Совместная энтропия Y и Z может быть записана как

$$H(Y,Z) = H(Y) + H(Z \mid Y) = H(Z) + H(Y \mid Z).$$

Из схемы понятно, что для заданного Y, то есть, когда выходное значение выбрано, значение Z также известно, поэтому

$$H(Z \mid Y) = 0$$

и

$$H(Y,Z) = H(Y) = H(Z) + H(Y \mid Z).$$

Тогда неопределённость Y может быть записана как

$$H(Y) = H(Y,Z) = H(Z) + H(Y \mid Z).$$

Из определения условной энтропии можно записать её следующим образом

$$H(Y) = H(Z) + H(Y \mid Z) = H(Z) + \sum_{i=1}^{L} H(Y \mid Z = i) P_Z(i) =$$

$$= H(Z) + \sum_{i=1}^{L} H\left(Y^{(i)}\right) q_i .$$

Последнее выражение следует из того факта, что при выбранном Z неопределённость Y связана с неопределённостью её выбора в соответствующем канале.

Аналогичным образом можно записать условную энтропию H(YZ\X) и точно так же H(Z\XY) равна нулю, поскольку для известного Y точно известна Z.

$$H(Y,Z \mid X) = H(Y \mid X) = H(Z \mid X) + H(Y \mid XZ).$$

Кроме того, Z и X статистически независимы, поэтому

$$H(Y \mid X) = H(Z) + H(Y \mid XZ).$$

Раскрыв запись H(Y\XZ), получим

$$H(Y \mid X) = H(Z) + \sum_{i=1}^{L} H(Y \mid X, Z = i) P_Z(i) = H(Z) + \sum_{i=1}^{L} H\left(Y^{(i)} \mid X\right) q_i.$$

Теперь можно определить количество информации как

$$I(X,Y) = H(Y) - H(Y \mid X).$$

Подставив соответствующие выражения для H(Y) и H(Y\X), получим

$$I(X,Y) = \sum_{i=1}^{L} \left[H\left(Y^{(i)}\right) - H\left(Y^{(i)} \mid X\right) \right] q_i .$$

И, по определению количества информации, выражение в скобках превращается в

$$I(X,Y) = \sum_{i=1}^{L} I\left(X, Y^{(i)}\right) q_i .$$

Непрерывный гауссов канал

Один из важнейших непрерывных каналов – Гауссов канал. Рассмотрим пропускную способность гауссова канала и докажем достижимость этой пропускной способности.

Гауссов канал, как правило представляется в виде суммы полезного сигнала X и шума Z. Предполагается, что распределения сигнала и шума независимы.

Пропускная способность гауссова канала с ограничением на мощность входного сигнала P и дисперсию шума N задаётся как

$$C = \max_{MX^2 \le P} I(X,Y) = \frac{1}{2}\log\left(1 + \frac{P}{N}\right),$$

где максимум достигается при $X \sim N(0,K)$.

Доказательство:

Пропускная способность задаётся выражением

$$C = \max_{p(x):MX^2 \le P} I(X,Y),$$

где максимизация производится по всем возможным распределениям входного сигнала p(x)

Из определения количества информации можно записать

$$\begin{aligned} I(X,Y) &= h(Y) - h(Y \mid X) = \\ &= h(Y) - h(X + Z \mid X) \\ &= h(Y) - h(Z \mid X) \qquad (*) \\ &= h(Y) - h(Z), \qquad (**) \end{aligned}$$

* – для заданного X неопределённость $X + Z$ равна неопределённости Z,

** – поскольку X и Z независимы.

Как известно, Z – нормально распределено, и поскольку дифференциальная энтропия $Z \sim N(0,K)$ вычисляется как

$$h(Z) = \frac{1}{2}\log_2(2\pi e N).$$

А также известно, что

$$MY^2 = M(X+Z)^2 = MX^2 + 2 \cdot MX \cdot MZ + MZ^2 = P + N.$$

Дифференциальная энтропия Y с дисперсией $P + N$ ограничена сверху дифференциальной энтропией гауссовой случайной величины

$$
\begin{aligned}
I(X,Y) &= h(Y) - h(Z) \\
&\le \frac{1}{2}\log_2\left(2\pi e(P+N)\right) - \frac{1}{2}\log_2\left(2\pi eN\right) \\
&= \frac{1}{2}\log_2\left(1+\frac{P}{N}\right).
\end{aligned}
$$

Максимум достигается при $X \sim N(0,P)$, когда Y – гауссова случайная величина и равенство выполняется.

Теперь покажем, что пропускная способность гауссова канала с ограничением по мощности сигнала и дисперсии шума равная

$C = \frac{1}{2}\log_2\left(1+\frac{P}{N}\right)$ бит за одну передачу достижима.

Тема 5. Символьные коды. Префиксные коды

Кодирование переменной длины. Префиксные коды

В этой лекции мы начнём новую тему, посвящённую эффективному кодированию источника, также известному как сжатие. В общем случае, все рассматриваемые коды могут быть использованы в любой системе счисления с целым показателем большим единицы. Поскольку, наиболее распространённой является система счисления с основанием равным двум, в настоящем пособии будут рассматриваться бинарные коды.

В первую очередь мы обсудим коды переменной длины, которые кодируют один символ источника за раз, вместо кодирования строк из N символов источника. Эти коды – без потерь. Они гарантируют сжатие и восстановление без ошибок, однако существует вероятность того, что закодированная строка окажется длиннее, чем исходная.

Идея, позволяющая осуществить сжатие, в общем, заключается в том, чтобы задать более короткие последовательности символов более вероятным исходам, а более длинные – менее вероятным.

Рассмотрим три основных требования к полезному коду. Во-первых, любая закодированная строка должна быть однозначно декодируемой. Во-вторых, символьный код должен быть прост для декодирования, В-третьих, код должен обеспечивать максимально возможное сжатие.

Таблица 3. Примеры символьных кодов

Символ	Код 1	Код 2	Код 3	Код 4	Код 5
a	00	00	0	1	0
b	00	01	1	10	10
c	01	10	11	100	110
d	11	11	100	1000	111

Любая закодированная строка должна быть однозначно декодируемой

Код называется однозначно декодируемым, если любая конечная последовательность кода соответствует не более, чем одному сообщению. Коды 2, 4, 5, приведённые в таблице – примеры однозначно декодируемого кода.

Символьный код должен быть прост для декодирования

Наиболее прост для декодирования код, в котором конец кодового слова может быть обнаружен одновременно с получением соответствующего символа, что может произойти в случае, когда никакое кодовое слово не является префиксом другого. бинарная последовательность z является префиксом другой бинарной последовательности z', если z имеет длину n и первые n знаков z' в точности составляют последовательность z.

Символьный код, в котором никакое кодовое слово не является префиксом для другого кодового слова, называется префиксным.

Префиксный код также известен как «мгновенно декодируемый» или «саморазделимый», поскольку закодированная строка может быть раскодирована слева направо без получения последующих кодовых слов. Конец кодового слова обнаруживается мгновенно. Префиксные коды *однозначно декодируемы*.

Коды 2 и 5, приведённые в таблице – префиксные.

Для декодирования префиксного кода строится бинарное дерево. Ниже приведён пример такого дерева для бинарного префиксного кода {011, 10, 11, 00}.

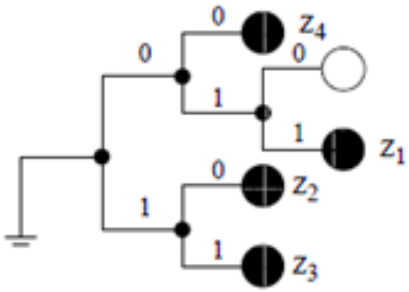

Рис. 9. Бинарное дерево для префиксного кода

Код должен обеспечивать максимально возможное сжатие

Средняя длина L(X) символьного для ансамбля X

$L(X) = \sum_x \left(P(x) l(x) \right)$.

Пример

Таблица 4. Символьный код с вероятностями

Символ	Вероятности	Код
a	½	0
b	¼	10
c	1/8	110
d	1/8	111

Энтропия $H(X) = 1,75$, а ожидаемая длина $L(X)$ также равна 1,75.

Последовательность acdbac кодируется как 0110111100110.

Найдём предел кодирования для однозначно декодируемых кодов.

Для начала рассмотрим код {00, 01, 10, 11}.

- Уменьшим длину одного кодового слова $00 \to 0$. Однозначная декодируемость может быть восстановлена только посредством увеличения длины других кодовых слов.
- Добавим новое кодовое слово: 110. Однозначная декодируемость может быть восстановлена только при увеличении длины одного из кодовых слов

Неравенство Крафта

Бинарный префиксный код, который содержит кодовые слова с длинами равными $l_1, l_2, ..., l_K$ (положительные целые) существует тогда и только тогда, когда

$\sum_{i=1}^{K} \left(2^{-l_i} \right) \le 1$.

Доказательство:

Пусть $S = \sum_i D^{-l_i}$.

Рассмотрим количество

$$S^N = \left[\sum_i 2^{-l_i}\right]^N = \sum_{i_1=1}^{I}\sum_{i_2=1}^{I}\ldots\sum_{i_N=1}^{I} 2^{-\left(l_{i_1}+l_{i_1}+\ldots+l_{i_N}\right)}.$$

Значение суммы в показателе степени $\left(l_{i_1}+l_{i_2}+\ldots+l_{i_N}\right)$ равно длине кодируемой строки $\mathbf{x}=a_{i_1}a_{i_2}\ldots a_{i_N}$. Для каждой строки $\mathbf{x}$ длины N там содержится одно слагаемое. Введём массив A_l, который посчитает, сколько строк $\mathbf{x}$ кодируется длиной l. Тогда определив $l_{\min}=\min_i l_i$ и $l_{\max}=\max_i l_i$:

$$S^N = \sum_{l=Nl_{\min}}^{Nl_{\max}} 2^{-l} A_l .$$

Теперь предположим, что код уникально декодируемый. Рассмотрим $\mathbf{x}$ с длиной кода l. Всего 2^l различных битовых строк длины l, поэтому должно выполняться неравенство $A_l \le 2^l$. Таким образом

$$S^N = \sum_{l=Nl_{\min}}^{Nl_{\max}} 2^{-l} A_l \le \sum_{l=Nl_{\min}}^{Nl_{\max}} 1 \le Nl_{\max} .$$

Следовательно, $S^N \le Nl_{\max}$ для всех N. Теперь, если S больше 1, тогда при увеличении N, S^N растёт экспоненциально и для достаточно больших N экспонента всегда превосходит полином, такой как $Nl_{\max}$. Но наш результат ($S^N \le Nl_{\max}$) должен быть истинным для любого N. Следовательно, $S \le 1$.

Для бинарного кода неравенство Крафта имеет вид

$$\sum_{i=1}^{K}\left(2^{-l_i}\right) \le 1.$$

Наилучшее достижимое сжатие

Ожидаемая длина уникально декодируемого кода ограничена снизу энтропией H(X).

Доказательство:

Определим вероятности $q_i \equiv 2^{-l_i} / z$, где $z = \sum_{i'} 2^{-l_{i'}}$, таким образом

$$l_i = \log \frac{1}{q_i} - \log z .$$

Используем неравенство Гиббса

$\sum_i p_i \log \frac{1}{q_i} \geq \sum_i p_i \log \frac{1}{p_i}$ с равенством только при $q_i = p_i$ и неравенство Крафта:

$$L(X) = \sum_i p_i l_i = \sum_i p_i \log \frac{1}{q_i} - \log z \geq$$

$$\geq \sum_i p_i \log \frac{1}{p_i} - \log z \geq H(X).$$

Равенство $L(X) = H(X)$ выполняется тогда и только тогда, когда неравенство Крафта становится равенством $z = 1$ и длины кодовых слов удовлетворяют $l_i = \log \frac{1}{p_i}$

Таким образом, сжать информацию меньше энтропии невозможно. Как близко можно приблизиться к энтропии?

Теорема о кодировании источника.

Для ансамбля X существует префиксный код C со средней длиной удовлетворяющей неравенству:

$$L(C, X) \in [H(X), H(X) + 1]$$

Доказательство:

Зададим длины кодовых слов равными целым числам немного большими, чем оптимальные длины

$$l_i = \left\lceil \log_2 \frac{1}{p_i} \right\rceil,$$

где $\lceil l^* \rceil$ обозначает наименьшее целое число, превышающее или равное l^*.

Проверим, что существует префиксный код с такими длинами, проверив неравенство Крафта.

$$\sum_i 2^{-l_i} = \sum_i 2^{-\lceil \log_2(1/p_i) \rceil} \le \sum_i 2^{-\log_2(1/p_i)} = \sum_i p_i = 1.$$

Затем подтвердим

$$L(X) = \sum_i 2^{-p_i \lceil \log_2(1/p_i) \rceil} < \sum_i 2^{-p_i (\log_2(1/p_i)+1)} = H(X) + 1.$$

Тема 6. Кодирование Шеннона-Фано. Кодирование Хаффмена. Арифметическое кодирование

Оптимальное кодирование Шеннона

В кодировании Шеннона символы располагаются в порядке от наиболее вероятных к наименее вероятным. Им присваиваются коды, путём взятия первых $l_i = \lceil \log_2 p_i \rceil$ цифр из двоичного разложения кумулятивной вероятности $\sum_{k=1}^{i-1} p_i$. Пример кодирования приведён в таблице 5. Кодирование Шеннона обеспечивает H(X) + 2.

Таблица 5. Кодирование Шеннона

a_i	$p(a_i)$	l_i	Сумма p_i до i-1	Сумма по $p(a_i)$	Итоговый код
a_1	0,37	2	0,0	0,0000	00
a_2	0,18	3	0,37	0,0101	010
a_3	0,18	3	0,55	0,1000	100
a_4	0,12	4	0,73	0,1011	1011
a_5	0,09	4	0,85	0,1101	1101
a_6	0,06	4	0,94	0,1111	1111

Оптимальное кодирование Шеннона-Фано

Код строится следующим образом. Кодируемые знаки выписывают в таблицу в порядке убывания их вероятностей в сообщениях. Затем их разделяют на две группы так, чтобы значения сумм вероятностей в каждой группе были близкими. Все знаки одной из групп в соответствующем разряде кодируются, например, единицей, тогда знаки второй группы кодируются нулем. Каждую полученную в процессе деления группу подвергают вышеописанной операции до тех пор, пока в результате очередного деления в каждой группе не останется по одному знаку. Пример кодирования приведён в таблице 6.

Таблица 6. Кодирование Шеннона-Фано

ai	$p(a_i)$	Процесс кодирования				Итоговый код
a_1	0,37	0	0			00
a_2	0,18		1			01
a_3	0,18	1	0			10
a_4	0,12		1	0		110
a_5	0,09			1	0	1110
a_6	0,06				1	1111

Оптимальное кодирование Хаффмена

1. Взять два наименее вероятных символа в алфавите. Эти два символа получат кодовые слова с максимальной длиной, отличающиеся последним символом.

2. Объединить два символа в один, повторить 1.

Таблица 7. Кодирование Хаффмена

a_i	$p(a_i)$	Построение дерева вероятностей					Итоговый код
a_1	0,37	0,37	0,37	0,37	0,63	1	0
a_2	0,18	0,18	0,27	0,36	0,37		111
a_3	0,18	0,18	0,18	0,27			110
a_4	0,12	0,15	0,18				100
a_5	0,09	0,12					1011
a_6	0,06						1010

Кодируемые знаки, также как при использовании метода Шеннона-Фано, располагают в порядке убывания их вероятностей (таблица 7). Далее на каждом этапе две последние позиции списка заменяются одной и ей приписывают вероятность, равную сумме вероятностей заменяемых позиций. После этого производится пересортировка списка по убыванию вероятностей, с сохранением информации о том, какие именно знаки объединялись на каждом этапе. Процесс продолжается до тех пор, пока не останется единственная позиция с вероятностью, равной 1.

После этого строится кодовое дерево. Корню дерева ставится в соответствие узел с вероятностью, равной 1. Далее каждому узлу

приписываются два потомка с вероятностями, которые участвовали в формировании значения вероятности обрабатываемого узла. Так продолжают до достижения узлов, соответствующих вероятностям исходных знаков.

Процесс кодирования по кодовому дереву осуществляется следующим образом. Одной из ветвей, выходящей из каждого узла, например, с более высокой вероятностью, ставится в соответствие символ 1, а с меньшей – 0. Спуск от корня к нужному знаку дает код этого знака. Правило кодирования в случае равных вероятностей оговаривается особо. Таблица 7 и рисунок 10 иллюстрируют применение методики Хаффмана.

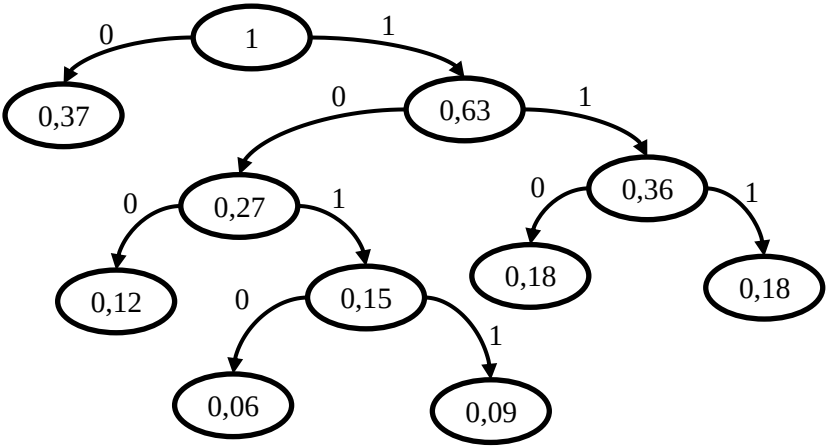

Рис. 10. Бинарное дерево кода Хаффмена

Блочное кодирование

Если величина среднего числа символов на знак оказывается значительно большей, чем энтропия, то это говорит об избыточности кода. Эту избыточность можно устранить, если перейти к кодированию блоками. Рассмотрим простой пример кодирования двумя знаками z_1, z_2 с вероятностями их появления в сообщениях 0,1 и 0,9 соответственно.

Если один из этих знаков кодировать, например, нулем, а другой единицей, т.е. по одному символу на знак, имеем соответственно

$l_{ср} = 0{,}1 \cdot 1 + 0{,}9 \cdot 1 = 1{,}0, \qquad H(z) = -0{,}1 \cdot \log_2 0{,}1 - 0{,}9 \cdot \log_2 0{,}9 = 0{,}47.$

При переходе к кодированию блоками по два знака (таблица 8)

$$l_{ср} = \frac{l_{ср,бл}}{2} = \frac{1}{2}\left(0{,}81 \cdot 1 + 0{,}09 \cdot 2 + 0{,}09 \cdot 3 + 0{,}01 \cdot 3\right) = 0{,}645.$$

Можно проверить, что при кодировании блоками по три символа среднее число символов на знак уменьшается и оказывается равным около 0,53. Эффект достигается за счет того, что при укрупнении блоков, группы можно делить на более близкие по значениям суммарных вероятностей подгруппы. Вообще $\lim\limits_{n\to\infty} l_{ср} = H(z)$, где n – число символов в блоке.

Таблица 8. Блочное кодирование

Блоки	Вероятности	Коды
$z_1\ z_1$	0,81	1
$z_1\ z_2$	0,09	01
$z_2\ z_1$	0,09	001
$z_2\ z_2$	0,01	000

Арифметическое кодирование

Арифметическое кодирование (англ. Arithmetic coding) — алгоритм сжатия информации без потерь, который при кодировании ставит в соответствие тексту вещественное число из отрезка [0; 1). Данный метод, как и алгоритм Хаффмана, является энтропийным, т.е. длина кода конкретного символа зависит от частоты встречаемости этого символа в тексте. Арифметическое кодирование показывает более высокие результаты сжатия, чем алгоритм Хаффмана, для данных с неравномерными распределениями вероятностей кодируемых символов. Кроме того, при арифметическом кодировании каждый символ кодируется нецелым числом бит, что эффективнее кода Хаффмана (теоретически, символу a с вероятностью появления p(a) допустимо ставить в соответствие код

длины $-\log_2 p(a)$, следовательно, при кодировании алгоритмом Хаффмана это достигается только с вероятностями, равными обратным степеням двойки).

Кодирование

На вход алгоритму передаются текст для кодирования и список частот встречаемости символов.

1. Рассмотрим отрезок [0; 1) на координатной прямой.
2. Поставим каждому символу текста в соответствие отрезок, длина которого равна частоте его появления.
3. Считаем символ из входного потока и рассмотрим отрезок, соответствующий этому символу. Разделим этот отрезок на части, пропорциональные частотам встречаемости символов.
4. Повторим пункт (3) до конца входного потока.
5. Выберем любое число из получившегося отрезка, которое и будет результатом арифметического кодирования.

Замечание: для оптимизации размера кода можно выбрать из полученного на последнем шаге диапазона [left; right] число, содержащее наименьшее количество знаков в двоичной записи.

Рассмотрим в качестве примера строку abacaba:

Таблица 9.

Символ	Частота появления
a	0,571429
b	0,285714
c	0,142857

Таблица 10

Считанный символ	Левая граница отрезка	Правая граница отрезка
	0	1
a	0	0,571429
b	0,326531	0,489796
a	0,326531	0,419825
c	0,406497	0,419825
a	0,406497	0,414113
b	0,410849	0,413025
a	0,410849	0,412093

Декодирование

Алгоритм по вещественному числу восстанавливает исходный текст.

1. Выберем на отрезке [0; 1), разделенном на части, длины которых равны вероятностям появления символов в тексте, подотрезок, содержащий входное вещественное число. Символ, соответствующий этому подотрезку, дописываем в ответ.
2. Нормируем подотрезок и вещественное число.
3. Повторим пункты 1—2 до тех пор, пока не получим ответ.

Тема 7. Другие эффективные коды

Коды Элиаса

Гамма-код

Гамма-код Элиаса — это универсальный код для кодирования положительных целых чисел, разработанный Питером Элиасом. Он обычно используется при кодировании целых чисел, максимальное значение которых не может быть определено заранее.

Алгоритм кодирования гамма-кодом Элиаса:

1. Записать число в двоичном представлении.
2. Перед двоичным представлением дописать нули, количество нулей на единицу меньше количества битов двоичного представления числа.

Алгоритм декодирования гамма-кода Элиаса

1. Считать все нули, встречающиеся до первой единицы. Пусть N — количество этих нулей.
2. Считать N+1 цифр целого числа.

Таблица 11

Число	Двоичное представление	Количество битов	Гамма-код Элиаса
1	1	1	1
2	10	2	0 10
3	11	2	0 11
4	100	3	00 100
5	101	3	00 101
6	110	3	00 110
7	111	3	00 111
8	1000	4	000 1000
9	1001	4	000 1001
10	1010	4	000 1010
11	1011	4	000 1011
12	1100	4	000 1100

Дельта-код

Дельта-код Элиаса — это модификация гамма-кода Элиаса, в котором число разрядов двоичного представления числа, в свою очередь, тоже кодируется дельта-кодом Элиаса.

Алгоритм кодирования дельта-кодом Элиаса:

1. Записать число без в двоичном представлении без старшей единицы.
2. Перед двоичным представлением записать количество битов двоичного представления *исходного* числа дельта-кодом Элиаса.

Алгоритм декодирования дельта -кода Элиаса

1. Считать все нули, встречающиеся до первой единицы. Пусть M — количество этих нулей.
2. Записать в L число, представленное следующими M+1 битов.
3. Считать следующие L битовых цифр и приписать к ним слева единицу.

Таблица 12.

Число	Двоичное представление	Количество битов	Гамма-код Элиаса числа битов	Дельта-код Элиаса
1	1	1	1	1
2	1 0	2	0 10	010 0
3	1 1	2	0 10	010 1
4	1 00	3	0 11	011 00
5	1 01	3	0 11	011 01
6	1 10	3	0 11	011 10
7	1 11	3	0 11	011 11
8	1 000	4	00 100	00100 001
9	1 001	4	00 100	00100 010
10	1 010	4	00 100	00100 011
11	1 011	4	00 100	00100 100
12	1 100	4	00 100	00100 101

Омега-код (рекурсивный код) Элиаса

Так же, как гамма- и дельта-код Элиаса, он приписывает к началу целого числа порядок его величины в универсальном коде. Однако, в отличие от двух других указанных кодов, омега-код рекурсивно кодирует префикс, именно поэтому он также известен, как рекурсивный код Элиаса.

Алгоритм кодирования омега-кодом Элиаса:

1. Переписать группу нолей в конец представления.
2. Если число, которое требуется закодировать, — единица, стоп; если нет, добавить двоичное представление числа в качестве группы в начало представления.
3. Повторить предыдущий шаг, с количеством только что записанных цифр(бит), минус один, как с новым числом, которое следует закодировать.

Алгоритм декодирования омега -кода Элиаса

1. Начать с переменной N, установленной в значение 1.
2. Считать первую «группу», следующую за остальными N разрядами, которая будет состоять либо из «0», либо из «1». Если она состоит из «0», это значит, что значение целого числа равно 1; если она начинается с «1», тогда N получает значение группы, которое интерпретируется как двоичное число.
3. Считывать каждую следующую группу; она будет состоять либо из «0», либо из «1», следующих за остальными N разрядами. Если группа равна «0», это значит, что значение целого числа равно N; если она начинается с «1», то N приобретает значение группы, интерпретируемой как двоичное число.

Таблица 13

Число	Омега-код Элиаса	Шаг 1	Шаг 2	Шаг 3
1	0	0		
2	10 0	0	10 0	
3	11 0	0	11 0	
4	10 100 0	0	100 0	10 100 0
5	10 101 0	0	101 0	10 101 0
6	10 110 0	0	110 0	10 110 0
7	10 111 0	0	111 0	10 111 0
8	11 1000 0	0	1000 0	11 1000 0
9	11 1001 0	0	1001 0	11 1001 0
10	11 1010 0	0	1010 0	11 1010 0
11	11 1011 0	0	1011 0	11 1011 0
12	11 1100 0	0	1100 0	11 1100 0

Словарные коды

Алгоритм LZW

Алгоритм Лемпеля — Зива — Велча (Lempel-Ziv-Welch, LZW) — это универсальный алгоритм сжатия данных без потерь.

Непосредственным предшественником LZW является алгоритм LZ78, опубликованный Абрахамом Лемпелем(Abraham Lempel) и Якобом Зивом (Jacob Ziv) в 1978 г. Этот алгоритм воспринимался как математическая абстракция до 1984 г., когда Терри Уэлч (Terry A. Welch) опубликовал свою работу с модифицированным алгоритмом, получившим в дальнейшем название LZW (Lempel-Ziv-Welch).

Описание

Процесс сжатия выглядит следующим образом. Последовательно считываются символы входного потока и происходит проверка, существует ли в созданной таблице строк такая строка. Если такая строка существует, считывается следующий символ, а если строка не существует, в поток заносится код для предыдущей найденной строки, строка заносится в таблицу, а поиск начинается снова.

Например, если сжимают байтовые данные (текст), то строк в таблице окажется 256 (от «0» до «255»). Если используется 10-битный код, то под коды для строк остаются значения в диапазоне от

256 до 1023. Новые строки формируют таблицу последовательно, т. е. можно считать индекс строки ее кодом.

Алгоритму декодирования на входе требуется только закодированный текст, поскольку он может воссоздать соответствующую таблицу преобразования непосредственно по закодированному тексту. Алгоритм генерирует однозначно декодируемый код за счет того, что каждый раз, когда генерируется новый код, новая строка добавляется в таблицу строк. LZW постоянно проверяет, является ли строка уже известной, и, если так, выводит существующий код без генерации нового. Таким образом, каждая строка будет храниться в единственном экземпляре и иметь свой уникальный номер. Следовательно, при дешифровании при получении нового кода генерируется новая строка, а при получении уже известного, строка извлекается из словаря.

Псевдокод алгоритма

1. Инициализация словаря всеми возможными односимвольными фразами. Инициализация входной фразы ω первым символом сообщения.
2. Считать очередной символ K из кодируемого сообщения.
3. Если КОНЕЦ_СООБЩЕНИЯ, то выдать код для ω, иначе:
4. Если фраза ω(K) уже есть в словаре, присвоить входной фразе значение ω(K) и перейти к Шагу 2, иначе выдать код ω, добавить ω(K) в словарь, присвоить входной фразе значение K и перейти к Шагу 2.
5. Конец.

Пример кодирования

Пусть мы сжимаем последовательность «abacabadabacabae».

Шаг 1: Тогда, согласно изложенному выше алгоритму, мы добавим к изначально пустой строке “a” и проверим, есть ли строка “a” в таблице. Поскольку мы при инициализации занесли в таблицу все строки из одного символа, то строка “a” есть в таблице.

Шаг 2: Далее мы читаем следующий символ «b» из входного потока и проверяем, есть ли строка “ab” в таблице. Такой строки в таблице пока нет.

Добавляем в таблицу <5> “ab”. В поток: <0>;

Шаг 3: “ba” — нет. В таблицу: <6> “ba”. В поток: <1>;

Шаг 4: “ac” — нет. В таблицу: <7> “ac”. В поток: <0>;

Шаг 5: “ca” — нет. В таблицу: <8> “ca”. В поток: <2>;

Шаг 6: “ab” — есть в таблице; “aba” — нет. В таблицу: <9> “aba”. В поток: <5>;

Шаг 7: “ad” — нет. В таблицу: <10> “ad”. В поток: <0>;

Шаг 8: “da” — нет. В таблицу: <11> “da”. В поток: <3>;

Шаг 9: “aba” — есть в таблице; “abac” — нет. В таблицу: <12> “abac”. В поток: <9>;

Шаг 10: “ca” — есть в таблице; “cab” — нет. В таблицу: <13> “cab”. В поток: <8>;

Шаг 11: “ba” — есть в таблице; “bae” — нет. В таблицу: <14> “bae”. В поток: <6>;

Шаг 12: И, наконец последняя строка “e”, за ней идет конец сообщения, поэтому мы просто выводим в поток <4>.

Таблица 14

Текущая строка	Текущий символ	Следующий символ	Вывод		Словарь
			Код	Биты	
ab	a	b	0	000	5: ab
ba	b	a	1	001	6: ba
ac	a	c	0	000	7: ac
ca	c	a	2	010	8: ca
ab	a	b	-	-	-
aba	b	a	5	101	9: aba
ad	a	d	0	000	10: ad
da	d	a	3	011	11: da
ab	a	b	-	-	-
aba	b	a	-	-	-
abac	a	c	9	1001	12: abac
ca	c	a	-	-	-
cab	a	b	8	1000	13: cab
ba	b	a	-	-	-
bae	a	e	6	0110	14: bae
e	e	-	4	0100	-

Итак, мы получаем закодированное сообщение «0 1 0 2 5 0 3 9 8 6 4», что на 11 бит короче.

Декодирование

Особенность LZW заключается в том, что для декомпрессии нам не надо сохранять таблицу строк в файл для распаковки. Алгоритм построен таким образом, что мы в состоянии восстановить таблицу строк, пользуясь только потоком кодов.

Теперь представим, что мы получили закодированное сообщение, приведённое выше, и нам нужно его декодировать. Прежде всего, нам нужно знать начальный словарь, а последующие записи словаря мы можем реконструировать уже на ходу, поскольку они являются просто конкатенацией предыдущих записей.

Таблица 15

Данные		На выходе	Новая запись	
Биты	Код		Полная	Частичная
000	0	a	-	5: a?
001	1	b	5: ab	6: b?
000	0	a	6: ba	7: a?
010	2	c	7: ac	8: c?
101	5	ab	8: ca	9: ab?
000	0	a	9: aba	10: a?
011	3	d	10: ad	11: d?
1001	9	aba	11: da	12: aba?
1000	8	ca	12: abac	13: ca?
0110	6	ba	13: cab	14: ba?
0100	4	e	14: bae	-

+ Не требует вычисления вероятностей встречаемости символов или кодов.

+ Для декомпрессии не надо сохранять таблицу строк в файл для распаковки. Алгоритм построен таким образом, что мы в состоянии восстановить таблицу строк, пользуясь только потоком кодов.

+ Данный тип компрессии не вносит искажений в исходный графический файл, и подходит для сжатия растровых данных любого типа.

- Алгоритм не проводит анализ входных данных, поэтому не оптимален.

Применение

Опубликование алгоритма LZW произвело большое впечатление на всех специалистов по сжатию информации. За этим последовало большое количество программ и приложений с различными вариантами этого метода.

Этот метод позволяет достичь одну из наилучших степеней сжатия среди других существующих методов сжатия графических данных, при полном отсутствии потерь или искажений в исходных файлах. В настоящее время испольуется в файлах формата TIFF, PDF, GIF, PostScript и других, а также отчасти во многих популярных программах сжатия данных (ZIP, ARJ, LHA).

Тема 8. Помехоустойчивое кодирование. Код Хэмминга

Основные характеристики помехоустойчивого кодирования

Избыточное (помехоустойчивое) кодирование (англ. redundant encoding) — вид кодирования, использующий избыточное количество информации с целью последующего контроля целостности данных при записи/воспроизведении информации или при её передаче по линиям связи.

В наиболее простом случае (кодирование постоянной длины) процедура кодирования заключается в сопоставлении k информационным символам, соответствующих кодируемому знаку, блока из n символов.

k – число информационных разрядов,

n – общее число разрядов в помехоустойчивом коде,

$n-k$ – число проверочных разрядов.

В этом случае код обозначается как (n,k)

Корректирующая способность кода характеризуется двумя значениями.

r – кратность обнаруживаемых ошибок. Это число разрядов, при одновременном возникновении ошибок в которых гарантируется только обнаружение факта наличия ошибки, при этом не гарантируется определение точных разрядов, в которых эти ошибки возникли.

s – кратность исправляемых ошибок. Это число разрядов, при одновременном возникновении ошибок в которых гарантируется не только обнаружение факта наличия ошибки, но и определение точных разрядов, в которых эти ошибки возникли.

Коды, в которых возможно автоматическое исправление ошибок, называются самокорректирующимися. В настоящее время наибольший интерес представляют двоичные блочные корректирующие коды. При использовании таких кодов информация передаётся в виде блоков одинаковой длины и каждый блок кодируется и декодируется независимо друг от друга. Почти во всех

блочных кодах символы можно разделить на информационные и проверочные. Таким образом, все комбинации кодов разделяются на разрешенные (для которых соотношение информационных и проверочных символов возможно) и запрещенные.

Коды повторения

В данном коде каждый передаваемый символ повторяется ровно n раз. Соответственно, обозначается этот код $(n,1)$.

Рассмотрим его корректирующую способность для различных n

Таблица 16

n	2	3	4
Исходное сообщение 1	11	111	1111
Сообщение с одной ошибкой	01	101	1101
Вывод по сообщению с одной ошибкой	Ошибка есть, разряд неизвестен	Ошибка есть, разряд 2	Ошибка есть, разряд 2
Сообщение с двумя ошибками	00	100	1100
Вывод по сообщению с двумя ошибками	Ошибки нет	Ошибка есть, разряд 3	Двойная ошибка, разряд неизвестен
Сообщение с тремя ошибками	-	000	1000
Вывод по сообщению с тремя ошибками	-	Ошибки нет	Ошибка есть, разряд 4

Рассмотрим ряд требований к длине помехоустойчивых кодов.

Разрешёнными комбинаций называют кодовые последовательности, не содержащие ошибок. Иначе говоря, это последовательности, которые могут появиться в результате кодирования.

Вектором ошибок называют двоичную последовательность, содержащую единицы в разрядах, подверженных ошибкам, и нули в остальных разрядах. Соответственно, любая искаженная комбинация

может рассматриваться как результат сложения по модулю 2 исходной разрешенной комбинации и вектора ошибки.

Очевидно, что число разрешённых последовательностей кода (n,k) равно 2^k.

Рассмотрим количество возможных кратных ошибок.

Число ошибок кратности 1 – C_n^1, кратности 2 – C_n^2, кратности i – C_n^i.

Для любого выбранного вектора ошибок может быть сформирована подгруппа из 2^k кодовых последовательностей, полученных из разрешённых посредством сложения с этим вектором (внесения этой ошибки). Предположим, что код (n,k) должен исправлять ошибки кратности до s включительно. Тогда упомянутые выше подгруппы для всех векторов ошибок кратности не выше s должны быть непересекающимися для того, чтобы существовала возможность исправления этих ошибок.

Общее число подгрупп равно:

$$1(\textit{разрешённые}) + C_n^1(\textit{кратность}\ 1) +$$

$$+C_n^2(\textit{кратность}\ 2) + \ldots + C_n^s(\textit{кратность}\ s)$$

Поскольку подгруппы не пересекаются, а в каждой по 2^k комбинаций, общее их число:

$$2^k\left(1 + C_n^1 + C_n^2 + \ldots + C_n^s\right).$$

Максимально число комбинаций, которые могут быть сформированы кодом (n,k) – 2^n.

$$2^k\left(1 + C_n^1 + C_n^2 + \ldots + C_n^s\right) \le 2^n.$$

После преобразований получаем нижнюю границу Хэмминга длины помехоустойчивого кода:

$$C_n^1 + C_n^2 + \ldots + C_n^s \le 2^{n-k} - 1$$

или

$$n \ge k + \log_2\left(1 + C_n^1 + C_n^2 + \ldots + C_n^s\right).$$

Связь корректирующей способности с кодовым расстоянием

Вес (Хэмминга) кодовой последовательности определяется как число ненулевых компонент этой последовательности. Ясно, что кодовое расстояние между двумя последовательностями равно весу некоторой третьей последовательности, являющейся их суммой, которая (в силу свойства операции сложения по модулю два) также обязана быть последовательностью данного кода. Следовательно, минимальное кодовое расстояние для линейного кода равно минимальному весу его ненулевых векторов.

Кодовое расстояние d выражается числом символов, в которых последовательности отличаются друг от друга. Для определения кодового расстояния между двумя комбинациями двоичного кода достаточно сложить их по модулю 2, и подсчитать число единиц в полученном результате. Минимальное расстояние, подсчитанное по всем парам разрешенных кодовых комбинаций, называют минимальным кодовым расстоянием данного кода.

Обычно декодирование осуществляется таким образом, что любая принятая запрещенная кодовая комбинация отождествляется с разрешенной комбинацией, находящейся от неё на минимальном кодовом расстоянии. Если минимальное кодовое расстояние данного кода $d=1$, т.е. все комбинации кода являются разрешенными, то обнаружить ошибку не удастся. Если $d=2$, то удастся обнаружить единичную ошибку и т.д. В общем случае при необходимости обнаружения ошибки кратности до r включительно, минимальное кодовое расстояние должно удовлетворять условию

$$d_{\min} \geq r+1.$$

Для исправления ошибок кратности s, в соответствии с описанной в разделе 7.3 общей схемой построения группового кода, каждой разрешенной кодовой комбинации необходимо поставить в соответствие подмножество запрещенных комбинаций так, чтобы эти подмножества не пересекались. Для этого должно выполняться неравенство

$$d_{\min} \geq 2s+1.$$

Для исправления ошибок кратности s и одновременного обнаружения всех ошибок кратности r ($r \geq s$) минимальное кодовое (хэммингово) расстояние должно удовлетворять неравенству

$$d_{\min} \geq r + s + 1.$$

Дадим геометрическую трактовку приведенным выше соотношениям.

Любая n-разрядная двоичная кодовая комбинация может быть интерпретирована как вершина n-мерного гиперкуба с длиной ребра равной 1. Например, при $n = 2$ это квадрат, при $n = 3$ – единичный куб. В общем случае n-мерный гиперкуб содержит 2^n вершин, что совпадает с возможным числом n-разрядных двоичных кодовых комбинаций.

Кодовое расстояние можно интерпретировать, как наименьшее число ребер, которое надо пройти, чтобы попасть из одной разрешенной комбинации в другую. В подмножество каждой разрешенной комбинации тносят все вершины, оказавшиеся в сфере радиуса

$$s \leq (d - 1)/2.$$

Если в результате действия шума разрешенная комбинация переходит в точку, принадлежащую сфере, то она может быть исправлена.

Кодирование Хэмминга

Коды Хэмминга являются самоконтролирующимися кодами, то есть кодами, позволяющими автоматически обнаруживать ошибки при передаче данных. Для их построения достаточно приписать к каждому слову один добавочный (контрольный) двоичный разряд и выбрать цифру этого разряда так, чтобы общее количество единиц в изображении любого числа было, например, нечетным. Одиночная ошибка в каком-либо разряде передаваемого слова (в том числе, может быть, и в контрольном разряде) изменит четность общего количества единиц. Счетчики по модулю 2, подсчитывающие

количество единиц, которые содержатся среди двоичных цифр числа, могут давать сигнал о наличии ошибок.

При этом невозможно узнать, в каком именно разряде произошла ошибка, и, следовательно, нет возможности исправить её. Остаются незамеченными также ошибки, возникающие одновременно в двух, четырёх, и т. д. — в четном количестве разрядов. Впрочем, двойные, а тем более четырёхкратные ошибки полагаются маловероятными.

Построение кодов Хэмминга основано на принципе проверки на четность числа единичных символов: к последовательности добавляется такой элемент, чтобы число единичных символов в получившейся последовательности было четным.

Следующий алгоритм генерирует код, исправляющий одиночные ошибки для любого количества битов.

1. Пронумеровать разряды начиная с 1: 1, 2, 3, 4, 5, ….

2. Записать номера разрядов в двоичном представлении: 1, 10, 11, 100, 101, ….

3. Все разряды, являющиеся степенями двойки – биты чётности: 1, 2, 4, 8, …. (1, 10, 100, 1000)

4. Все остальные разряды – информационные.

Каждый бит данных включается в уникальный набор из двух или более битов чётности в соответствии с бинарным представлением его порядкового номера.

Бит чётности 1 покрывает все битовые позиции, номер которых содержит единицу в младшем бите: 1, 3, 5, 7, 9, …

Бит чётности 2 покрывает все битовые позиции, номер которых содержит единицу во втором бите: 2, 3, 6, 7, 10, …

Бит чётности 4 покрывает все битовые позиции, номер которых содержит единицу во втором бите: 4–7, 12–15, …

Бит чётности 8 покрывает все битовые позиции, номер которых содержит единицу во втором бите: 8–15, 24–31, …

В общем, каждый бит чётности покрывает все битовые позиции, для номера которых побитовое И с номером этого бита не равен нулю.

Это правило может быть представлено визуально:

Таблица 17

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
	ч1	ч2	и1	ч4	и2	и3	и4	ч8	и5	и6	и7	и8	и9	и10	и11	ч16	и12	и13	и14	и15	
ч1	1		1		1		1		1		1		1		1		1		1		
ч2		1	1			1	1			1	1			1	1			1	1		…
ч4				1	1	1	1					1	1	1	1					1	
ч8								1	1	1	1	1	1	1	1						
ч16																1	1	1	1	1	

Показаны только 20 закодированных битов (5 – чётности, 15 – информационных), но шаблон можно продолжать бесконечно. Ключевой особенностью кода Хэмминга, которая видна из примера, является то, что каждый выбранный бит входит в уникальный набор битов чётности. Для проверки на ошибки проверяются все биты чётности. Шаблон ошибок, называемый синдромом показывает бит ошибки. Если все биты чётности верны – ошибки нет, иначе, сумма позиций ошибочных битов чётности покажет бит с ошибкой. Если ошибочен ровно один бит чётности – ошибка в нём.

Как можно заметить, если имеется m битов чётности, они могут покрыть от 1 до 2^m-1 битов. Если вычесть биты чётности, остаётся 2^m – m – 1 информационных битов. Изменяя m, получим все возможные коды Хэмминга.

Таблица 18

Разряд		1	2	3	4	5	6	7	
Название бита		ч1	ч2	и1	ч4	и2	и3	и4	
Код		0	1	1	0	0	1	1	…
ч1	0	1	0	1	0	1	0	1	
ч2	1	0	1	1	0	0	1	1	
ч4	0	0	0	0	1	1	1	1	

В левой части таблицы мы оставили пустым один столбец, в который поместим результаты вычислений контрольных битов. Вычисление контрольных битов производим следующим образом. Берём одну из строк матрицы преобразования и находим её скалярное

произведение с кодовым словом, то есть перемножаем соответствующие биты обеих строк и находим сумму произведений. Если произведение получилось больше единицы, находим остаток от его деления на 2. Иными словами, мы подсчитываем сколько раз в кодовом слове и соответствующей строке матрицы в одинаковых позициях стоят единицы и берём это число по модулю 2.

Код Хэмминга с дополнительной проверкой на чётность (Исправление одиночной ошибки, обнаружение двойной ошибки)

Коды Хэмминга имеют минимальное расстояние равное трём. Это означает, что эти коды могут обнаруживать и исправлять одиночную ошибку, но не могут отличить двойную ошибку от одиночной.

Таким образом, такой код может обнаруживать двойные ошибки, если он не используется для исправления ошибок.

Для исправления этого недостатка, коды Хэмминга могут быть расширены дополнительным битом чётности. Таким образом можно увеличить минимальное кодовое расстояние до четырёх, что позволяет различать одиночные и двойные ошибки. Таким образом декодер может обнаруживать и исправлять одинарные ошибки и обнаруживать (но не исправлять) двойные. Если такой код не использовать для исправления ошибок, он может обнаруживать тройные ошибки.

Такой расширенный код Хэмминга используется в системах компьютерной памяти, где они известен как SECDED (аббревиатура Single Error Correction, Double Error Detection). В частности, используются коды (72,64), (127,120).

Алгоритм декодирования

Алгоритм декодирования по Хэммингу абсолютно идентичен алгоритму кодирования. Матрица преобразования соответствующей размерности умножается на матрицу-столбец кодового слова и каждый элемент полученной матрицы-столбца берётся по модулю 2. Полученная матрица-столбец получила название «матрица синдромов». Легко проверить, что кодовое слово, сформированное в соответствии с алгоритмом, описанным в предыдущем разделе, всегда даёт нулевую матрицу синдромов.

Матрица синдромов становится ненулевой, если в результате ошибки (например, при передаче слова по линии связи с шумами) один из битов исходного слова изменил своё значение. Предположим для примера, что в кодовом слове, полученном в предыдущем разделе, шестой бит изменил своё значение с нуля на единицу (на рисунке обозначено красным цветом). Тогда получим следующую матрицу синдромов.

Таблица 19

Разряд		1	2	3	4	5	6	7	
Название бита		ч1	ч2	и1	ч4	и2	и3	и4	
Код		0	1	0	0	0	1	1	…
с1	1	1	0	1	0	1	0	1	
с2	1	0	1	1	0	0	1	1	
с4	0	0	0	0	1	1	1	1	

Тема 9. Циклические коды

Операции на многочленах

Описание циклических кодов удобно проводить с помощью многочленов. Для этого вводят фиктивную переменную x, степени которой соответствуют номерам разрядов, начиная с 0. В качестве коэффициентов многочленов берут цифры 0 и 1, т.е. вводятся в рассмотрение многочлены над полем $\mathbf{GF}(2)$. Например, первая строка 1001011 описывается многочленом

$1 \cdot x^6 + 0 \cdot x^5 + 0 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x^1 + 1 \cdot x^0 = x^3 + x + 1$.

Многочлен для каждой следующей строки образуется путем умножения на x. При этом, если крайний левый символ отличается от нуля для реализации операции переноса единицы в конец комбинации из результата необходимо вычесть (сложить по модулю 2) многочлен $x^n + 1$.

Все комбинации циклического кода могут быть построены на кольце многочленов путем задания на множестве n-разрядных кодовых комбинаций двух операций – сложения и умножения. Операция сложения многочленов в данном случае реализуется как сложение соответствующих коэффициентов по модулю 2.

Операция умножения реализуется в следующей последовательности. Многочлены перемножаются как обычно с последующим приведением коэффициентов по модулю 2. Если в результате умножения получается многочлен степени n и выше, то осуществляется его деление на заданный многочлен степени n, а результатом умножения считают остаток от деления. Ясно, что старшая степень этого остатка не будет превышать величины $n-1$, а полученный остаток будет соответствовать некоторой n-разрядной кодовой комбинации, т.е. обеспечивается замкнутость.

Операция деления является обычным делением многочленов, только вместо вычитания используется сложение по модулю 2.

$$
\begin{array}{l|l}
x^6+x^4+x^2+1 & x^3+x+1 \\
\underline{x^6+x^4+x^3} & \overline{x^3+1} \\
x^3+x^2+1 & \\
\underline{x^3+x+1} & \\
x^2+x &
\end{array}
$$

Для реализации циклического сдвига с использованием описанной операции умножения необходимо после умножения на x выполнить деление на двучлен x^n+1. Эта операция называется *взятием остатка* или *приведением по модулю* x^n+1, а сам остаток называют вычетом:

$$
\begin{array}{rcl|l}
(x^{n-1}+x^{n-2}+\ldots+x+1)\cdot x & = & x^n+x^{n-1}+\ldots+x^2+x & \underline{x^n+1} \\
 & \oplus & \underline{x^n+1} & 1 \\
 & & 0+x^{n-1}+\ldots+x^2+x+1 &
\end{array}.
$$

Нетрудно заметить, что в данном случае остаток (вычет) формируется путем сложения по модулю 2 двучлена x^n+1 с результатом умножения на x.

Понятие и общая схема построения циклического кода

Циклическим называется код, каждая комбинация которого может быть построена в виде линейной комбинации кодов, каждый из которых получается путем циклического сдвига некоторой базисной комбинации, принадлежащей этому же коду. Если сдвиг осуществляется справа налево, крайний левый символ переносится в конец кодовой комбинации.

Описание циклических кодов удобно проводить с помощью многочленов. Для этого вводят фиктивную переменную x, степени которой соответствуют номерам разрядов, начиная с 0. В качестве коэффициентов многочленов берут цифры 0 и 1, т.е. вводятся в рассмотрение многочлены над полем $\mathbf{GF}(2)$. Например, первая строка 1001011 описывается многочленом

$1 \cdot x^6 + 0 \cdot x^5 + 0 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x^1 + 1 \cdot x^0 = x^3 + x + 1$.

Выделим в кольце подмножество всех многочленов, кратных некоторому многочлену $g(x)$. Ясно, что это подмножество будет идеалом, а многочлен $g(x)$ – *порождающим* или *образующим* многочленом идеала. Если $g(x) = 0$, то весь идеал состоит из одного этого многочлена. Если $g(x) = 1$, то в идеал войдут все многочлены кольца.

Известно [1], [9], что в кольце 2^n всех возможных многочленов степени n-1 над полем $\mathbf{GF}(2)$ неприводимый многочлен $g(x)$ степени $m = n - k$ порождает 2^k элементов идеала. Следовательно, циклический двоичный код можно определить как идеал, каждому многочлену которого ставится в соответствие n-разрядная разрешенная кодовая комбинация. Установим, каким требованиям при этом должен удовлетворять образующий многочлен идеала – $g(x)$.

Порождающий полином должен удовлетворять следующим требованиям:

- p(x) должен быть ненулевым;
- вес p(x) не должен быть меньше минимального кодового расстояния: v(p(x)) ≥ dmin;
- p(x) должен иметь максимальную степень k (k — число избыточных элементов в коде);
- p(x) должен быть делителем полинома ($x^n - 1$).

Если $g(x)$ удовлетворяет этому требованию, то кольцо многочленов можно разложить на классы вычетов по идеалу. Для наглядности схема разложения представлена в таблице 20. Первой строкой в этой таблице является сам идеал вместе с нулевым многочленом. В качестве образующих элементов классов берутся (соответствующие векторам ошибок) многочлены $r(x)$, не

принадлежащие идеалу, а классы вычетов по идеалу образуются путем сложения элементов идеала с образующими многочленами.

Таблица 20

0	$r_1(x)$	…	$r_z(x)$
$g(x)$	$g(x)+r_1(x)$	…	$g(x)+r_z(x)$
$xg(x)$	$xg(x)+r_1(x)$	…	$xg(x)+r_z(x)$
$(x+1)g(x)$	$(x+1)g(x)+r_1(x)$	…	$(x+1)g(x)+r_z(x)$
…	…	…	…
$f(x)\cdot g(x)$	$f(x)\cdot g(x)+r_1(x)$	…	$f(x)\cdot g(x)+r_z(x)$

Если реализована указанная схема образования классов вычетов, а многочлен $g(x)$ степени $m=n-k$ является делителем двучлена x^n+1, то каждый элемент кольца либо делится на $g(x)$ без остатка (тогда он элемент идеала), либо появляется остаток от деления $r(x)$ – это многочлен степени не выше $m-1$. Элементы кольца, дающие один и тот же остаток $r(x)$, относят к одному классу вычетов.

Таблица 21

М	Код	$g(x)$
1	11	$x+1$
2	111	x^2+x+1
3	1011	x^3+x+1
3	1101	x^3+x^2+1

Корректирующая способность кода тем выше, чем больше классов вычетов, т.е. остатков $r(x)$. Наибольшее число остатков 2^m-1 дает неприводимый многочлен. В качестве примера в таблице 8.3 приведены неприводимые многочлены до третьей степени

включительно. Таблицы, включающие большое число неприводимых многочленов, можно найти, например, в [2], [3].

Выбор образующих многочленов для обнаружения и исправления одиночных ошибок

Исправление одиночных ошибок. Каждой одиночной ошибке в одном из n разрядов должен соответствовать свой класс вычетов и свой опознаватель – остаток от деления на образующий многочлен $g(x)$. Как указывалось выше, наибольшее число остатков дает неприводимый многочлен. Если $m = n - k$ степень этого многочлена, число ненулевых остатков будет $2^{n-k} - 1$. Таким образом, для исправления всех n одиночных ошибок необходимо, чтобы выполнялось неравенство

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

Следовательно степень образующего многочлена

$$m = n - k \geq \log_2(n+1).$$

Выше было показано, что образующий многочлен должен быть делителем $x^n + 1$. С другой стороны, известно, что любой двучлен вида

$$x^{2^{m-1}} + 1 = x^n + 1$$

всегда может быть представлен в виде произведения всех неприводимых многочленов, степени которых являются делителями числа m от 1 до m включительно. Следовательно, для любого n существует хотя бы один неприводимый многочлен степени m, входящий сомножителем в разложение двучлена $x^n + 1$. Этот многочлен и может быть принят в качестве образующего.

Методы формирования комбинаций и декодирования циклического кода

Построение несистематического кода. Для построения n-разрядной разрешенной комбинации многочлен $a(x)$,

соответствующий кодируемой последовательности информационных символов, умножается на образующий многочлен:

$q(x) = a(x)g(x)$.

При декодировании (возможно отличающийся от $q(x)$) многочлен $\tilde{q}(x)$, соответствующий принятой комбинации, делят на $g(x)$. Ясно, что в случае отсутствия ошибок сразу получится исходный многочлен $a(x)$. Если в принятой комбинации содержится ошибка, при делении образуется остаток $r(x)$, т.е.

$\tilde{q}(x)/g(x) = f(x) + r(x)/g(x)$.

По остатку определяется класс вычетов и производится исправление ошибки.

Недостаток данного способа кодирования заключается в том, что после обнаружения и исправления ошибки необходимо снова делить на $g(x)$ для того, чтобы выделить информационные символы.

Построение систематического кода. Многочлен, соответствующий исходной информационной посылке $a(x)$, умножается на x^m. Образовавшиеся после умножения свободные младшие разряды заполняются остатком от деления данного выражения на образующий многочлен:

$q(x) = a(x) \cdot x^m + r(x)$.

Многочлен $q(x)$ обязан делиться на $g(x)$ без остатка. Покажем это.

При делении $a(x)x^m$ на $g(x)$ в общем случае имеем

$a(x) \cdot x^m / g(x) = c(x) + r(x)/g(x)$,

где $c(x)$ – целый полином. Это равенство (с учетом того, что операции вычитания и сложения по модулю два совпадают) можно переписать в виде

$a(x) \cdot x^m / g(x) + r(x)/g(x) = c(x)$,

или

$$q(x) = a(x) \cdot x^m + r(x) = c(x)g(x).$$

В данном случае информационные символы всегда остаются на первых k позициях. Такой код называют *систематическим*. При таком способе кодирования после исправления ошибок сразу становится известной исходная кодовая последовательность, занимающая первые k позиций.

Тема 10. Исправление пакетов ошибок. Циклический избыточный код

Исправление пакетов ошибок

Коды, которые мы ранее рассматривали, были спроектированы таким образом, чтобы исправлять случайные ошибки. В общем, код, исправляющий *t* ошибок, может исправлять любые шаблоны ошибок веса *t* или меньше в кодовом слове-блоке длины *n*. Однако, существуют каналы, в которых ошибки возникают небольшими интервалами, а не абсолютно случайно. К примеру, в средствах хранения данных ошибки возникают вследствие физических изменений, поэтому, скорее концентрированы, нежели случайно рассредоточены. Точно так же, помехи в короткие промежутки времени вызывают пакеты ошибок. Существует семейство кодов, используемых для исправления таких многократных ошибок. Рассмотрим их в этой лекции.

Назовём пакетом ошибок длины *t* вектор ошибок, ненулевые компоненты которого находятся в пределах *t* соседних разрядов.

Назовём циклическим пакетом ошибок длины *t* вектор ошибок, ненулевые компоненты которого находятся в пределах *t* соседних разрядов с учётом хотя бы в одной циклической перестановке этого вектора.

Примеры:

(0**1010110**000) – циклический пакет ошибок длины 6,

(000000**10001**) – циклический пакет ошибок длины 5,

(01000000**101**) – циклический пакет ошибок длины 5.

Можно описать пакет ошибок длины *t* в терминах полинома как

$$e(x) = x^i b(x) \quad \left(\mathrm{mod}\, x^n - 1\right)$$

где $b(x)$ полином степени *t-1*, описывающий шаблон ошибки, а *i* показывает, где начинается ошибка. Для предложенных выше примеров:

(0**1010110**000) $\qquad e(x) = \left(x^5 + x^3 + x + 1\right)x^4$

(0**1010110**000) $e(x)=\left(x^4+1\right)x^0$

(0**1010110**000) $e(x)=\left(x^4+x^2+1\right)x^9$

Рассмотрим линейный код C. Если все пакеты ошибок длины t или меньше возникают в различных подмножествах, тогда каждая может быть идентифицирована посредством её синдрома и все такие ошибки исправляемы. Более того, если C – линейный код, способный исправлять все пакеты ошибок длины t или меньше, тогда все такие ошибки должны возникать в различных подмножествах.

Предположим, что C может исправлять две таких различных ошибки e_1 и e_2, которые лежат в разных подмножествах c_i. Тогда $e_1-e_2=c$ – ненулевое кодовое слово. Предположим, что e_1 – полученный вектор. Как он может быть декодирован? Кодовое слово 0 может быть преобразовано в e_1 посредством внесения ошибки e_1 или кодовое слово c может быть преобразовано в e_1 посредством внесения ошибки e_2. Мы пришли к противоречию, поскольку этот код не способен исправлять пакеты ошибок длины *t* или меньше.

Улавливание ошибок

Циклический код может исправлять все пакеты ошибок длины *t* или меньше тогда и только тогда, если синдромы этих ошибок отличаются. Мы можем декодировать циклические пакеты ошибок посредством улавливания ошибок.

Можно доказать, что код (n, k) исправляющий пакеты ошибок длины t удовлетворяет ограничению $n-k\geq 2t$. Следовательно, $n-k\geq t$ и $n-t\geq k$. Теперь пакет ошибок длины t в кодовом слове длины n имеет циклическую последовательность из $n-t$ нулей, что является требованием для работы алгоритма улавливания ошибок. Мы приведём модификацию алгоритма улавливания ошибок, который может быть использован для всех пакетов ошибок длины t или меньше в циклическом коде исправления пакетов ошибок длины t.

(1) Вычислить синдром.

(2) Установить $i = 0$

(3) Если $s_i(x)$ нециклический пакет ошибок длины $\le t$, тогда $e(x) = x^{n-i}\left[s_i(x), 0\right]$.

(4) Пусть $I = I + 1$.

(5) Если $I = n$, остановиться, шаблон ошибок неопределим.

(6) Вычислить $s_i(x) = xs_{i-1}(x)$. Если степень $s_i(x) > n - k$, $s_i(x) = s_i(x) - g(x)$.

(7) Вернуться к шагу (3)

Пример

$g(x) = 1 + x + x^2 + x^3 + x^6$ (1001111) генерирует циклический код для исправления пакетов ошибок длины не более 3. Получен вектор (000.0011.0111.0111)

$r(x) = \left(x^2 + x^3\right)g(x) + \left(1 + x + x^4 + x^5\right)$.

Вычисляем синдромы

Таблица 22

i	Синдром
0	110011
1	101001
2	011101
3	111010
4	111011
5	111001
6	111101
7	110101
8	100101
9	000101

$e(x) = 10100000$

Некоторые подходящие многочлены

Удовлетворяющие условиям теоремы образующие многочлены сложно найти, поэтому приведём несколько примеров таких кодов для небольших t.

Таблица 23

g(x)	(n, k)	t
$1+x^2+x^3+x^4$	(7, 3)	2
$1+x^2+x^4+x^5$	(15, 10)	2
$1+x^4+x^5+x^6$	(31, 25)	2
$1+x^3+x^4+x^5+x^6$	(15, 9)	3
$1+x+x^2+x^3+x^6$	(15, 9)	3

Циклический избыточный код

Циклический избыточный код (англ. Cyclic redundancy check, CRC[1]) — алгоритм нахождения контрольной суммы, предназначенный для проверки целостности данных[2]. CRC является практическим приложением помехоустойчивого кодирования, основанном на определённых математических свойствах циклического кода.

Алгоритм CRC базируется на свойствах деления с остатком двоичных многочленов . Значение CRC является по сути остатком от деления многочлена, соответствующего входным данным, на некий фиксированный порождающий многочлен.

Параметры алгоритма

Одним из основных параметров CRC является порождающий полином.

С порождающим полиномом связан другой параметр — его степень, которая определяет количество битов, используемых для вычисления значения CRC. На практике наиболее распространены 8, 16- и 32-битовые слова, что является следствием особенностей архитектуры современной вычислительной техники.

Ещё одним параметром является начальное (стартовое) значение слова. Указанные параметры полностью определяют «традиционный» алгоритм вычисления CRC. Существуют также модификации алгоритма, например, использующие обратный порядок обработки битов.

Таблица 24

CRC-1		0x1
CRC-4-ITU		0x3
CRC-5-EPC		0x09
CRC-5-ITU		0x15
CRC-5-USB		0x05
CRC-6-CDMA2000-A		0x27
CRC-6-CDMA2000-B		0x07
CRC-6-DARC		0x19
CRC-6-ITU		0x03

Описание процедуры

Реализация CRC на логических элементах

К исходной строке добавляется строка из n нулей. Если старший бит в строку «1», то слово сдвигается влево на один разряд с последующим выполнением операции XOR с порождающим полиномом. Соответственно, если старший бит в слове «0», то после сдвига операция XOR не выполняется. Полученный после прохождения всей строки остаток и является контрольной суммой.

К исходной строке добавляется контрольная сумма из n битов. Если описанная выше процедура показывает в остатке 0, строка соответствует заданной контрольной сумме.

Тема 11. Матричные коды. Коды Адамара

Матричные коды

В теории кодирования образующей (порождающей) называется матрица, строки которой образуют базис линейного кода. Кодовыми словами в этом случае являются все линейные комбинации строк этой матрицы.

Если G – образующая матрица, кодовые слова помехоустойчивого кода формируются посредством следующего умножения.

$\mathbf{w} = \mathbf{sG}$,

где $\mathbf{s}$ – любой вектор.

Образующая матрица для кода (n, k) содержит k строк и n столбцов.

Стандартный вид образующей матрицы:

$\mathbf{G} = \left[\mathbf{I}_k \vdots \mathbf{P}_{k,n-k} \right]$.

Образующая матрица может быть использована для формирования проверочной матрицы (и наоборот):

$\mathbf{H} = \left[\mathbf{P}_{n-k,k}^T \vdots \mathbf{E}_{n-k} \right]$

Бинарные коды эквивалентны, если матрица одного кода может быть получена из другого следующими преобразованиями:

- перестановка столбцов
- перестановка строк
- сложение одной строки с другой

Построение матрицы-дополнения

Матрица-дополнение содержит всю информацию о схеме построения кода.

Существует формальный способ построения матрицы дополнения, основанный на следующем требовании. Вектор-строка, получающаяся в результате суммирования любых l, $(1 \le l \le k)$ строк матрицы дополнения, должна содержать не менее $d_{\min} - l$ отличных от нуля символов, где $d_{\min}$ – минимальное кодовое расстояние.

В соответствии с указанным требованием матрица-дополнение может строиться с соблюдением следующих правил:

- количество единиц в строке должно быть не менее $d_{\min} - 1$;

- сумма по модулю два двух любых строк должна содержать не менее $d_{\min} - 2$ единиц.

При соблюдении указанных требований комбинация, полученная суммированием любых 2-х строк образующей матрицы, будет содержать не менее $d_{\min}$ ненулевых символов.

Циклический код является групповым кодом, поэтому он может строиться с использованием матричных представлений так, как описано выше. Однако в данном случае появляются также некоторые дополнительные возможности, связанные со свойством цикличности. Рассмотрим способы построения образующей матрицы циклического кода.

Способ 1. Пусть образующий многочлен задан в виде

$g(x) = g_m x^m + ... + g_1 x + g_0$.

Тогда образующая матрица может быть построена путем умножения $g(x)$ на одночлен $x^{k-1}, k = n - m$ и последующим циклическим сдвигом так, что каждая i-я строка образующей матрицы составляется из коэффициентов многочлена , $g(x) \cdot x^{k-i}$ ($i = \overline{1,k}$):

$$\mathbf{M}_{n,k} = \begin{bmatrix} g_m & g_{m-1} & \cdots & g_0 & 0 & \cdots & 0 \\ 0 & g_m & g_{m-1} & \cdots & g_0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & g_m & g_{m-1} & \cdots & g_0 \end{bmatrix}.$$

Способ 2. Рассматриваются многочлены $Q_i(x)$, соответствующие коду, содержащему только один ненулевой разряд:

$Q_i(x) = x^{n-i}, i = \overline{1,k}$.

Для них вычисляются остатки

$r_i(x) = Q_i(x) / g(x)$.

Каждая i-я строка образующей матрицы формируется путем сложения по модулю два указанных многочленов и соответствующих им остатков. При этом образующая матрица (в данном случае систематического кода) представляется двумя подматрицами:

$$\mathbf{M}_{n,k} = \left[\mathbf{E}_k \vdots \mathbf{P}_{k,n-k}\right],$$

где $\mathbf{E}_k$ – единичная $k \times k$- матрица, а строками матрицы дополнения $\mathbf{P}_{k,n-k}$ являются остатки $r_i(x), i = \overline{1,k}$.

Коды Адамара

Рассмотрим правило формирования матрицы Адамара.

$$H_1 = [1],$$

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & - \end{bmatrix},$$

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & - & 1 & - \\ 1 & 1 & - & - \\ 1 & - & - & 1 \end{bmatrix},$$

$$H_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & - & 1 & - & 1 & - & 1 & - \\ 1 & 1 & - & - & 1 & 1 & - & - \\ 1 & - & - & 1 & 1 & - & - & 1 \\ 1 & 1 & 1 & 1 & - & - & - & - \\ 1 & - & 1 & - & - & 1 & - & 1 \\ 1 & 1 & - & - & - & - & 1 & 1 \\ 1 & - & - & 1 & - & 1 & 1 & - \end{bmatrix},$$

$$H_{2i} = \begin{bmatrix} H_i & H_i \\ H_i & -H_i \end{bmatrix},$$

Для кодирования и декодирования значению -1 матрицы Адамара (обозначается как "$-$") ставится в соответствие значение бита кода равное 0, а значению $+1$ матрицы Адамара (обозначается как "1") – значение бита кода равное 1.

Кодирование

Код Адамара предназначен для кодирования n символов входной последовательности в 2^n выходной. Для этого в качестве выходной последовательности берётся соответствующая строка матрицы Адамара.

$$c_j(x_i) = \begin{cases} \dfrac{H_{ij}+1}{2}, & 1 \le i \le n, \\ \dfrac{-H_{ij}+1}{2}, & n+1 \le i \le 2n, \end{cases}$$

$$\begin{bmatrix}0000\\0001\\0010\\0011\\0100\\0101\\0110\\0111\end{bmatrix}\rightarrow\begin{bmatrix}1&1&1&1&1&1&1&1\\1&-&1&-&1&-&1&-\\1&1&-&-&1&1&-&-\\1&-&-&1&1&-&-&1\\1&1&1&1&-&-&-&-\\1&-&1&-&-&1&-&1\\1&1&-&-&-&-&1&1\\1&-&-&1&-&1&1&-\end{bmatrix}\begin{bmatrix}1000\\1001\\1010\\1011\\1100\\1101\\1110\\1111\end{bmatrix}\rightarrow\begin{bmatrix}-&-&-&-&-&-&-&-\\-&1&-&1&-&1&-&1\\-&-&1&1&-&-&1&1\\-&1&1&-&-&1&1&-\\-&-&-&-&1&1&1&1\\-&1&-&1&1&-&1&-\\-&-&1&1&1&1&-&-\\-&1&1&-&1&-&-&1\end{bmatrix},$$

Декодирование

Умножим матрицу Адамара на полученную последовательность, получив вектор $\mathbf{F}$.

Определяем координату a (нумерация начинается с нуля), которой соответствует максимальное по модулю значение $\mathbf{F}$.

Если F_a отрицательна, то первая координата исходного сообщения равна 1, если F_a положительна – 0. Остальные координаты равны двоичному представлению a.

Пример 1

Пусть кодируется строка $\mathbf{x}=\begin{bmatrix}0&1&1&0\end{bmatrix}$.

После кодирования этой строке соответствует сообщение $\mathbf{c}=\begin{bmatrix}1&1&0&0&0&0&1&1\end{bmatrix}$.

Предположим, в ходе передачи возникла ошибка, в результате чего было принято сообщение $\mathbf{r}_1=\begin{bmatrix}1&0&0&0&0&0&1&1\end{bmatrix}$.

Запишем соответствующий этому сообщению вектор в нотации матрицы Адамара:

$\mathbf{v}=\begin{bmatrix}1&-&-&-&-&-&1&1\end{bmatrix}$.

После умножения матрицы Адамара на получившийся вектор, получаем вектор

$\mathbf{F}=\begin{bmatrix}-2&2&-2&2&-2&2&6&2\end{bmatrix}$.

Наименьшему значению вектора $\mathbf{F}$ соответствует координата $a=6$, при этом F_a положительна, следовательно исходное сообщение формируется как $\begin{bmatrix}0\vdots110\end{bmatrix}$.

Пример 2

Пусть кодируется строка $\mathbf{x} = [1 \quad 0 \quad 1 \quad 1]$.

После кодирования получаем $\mathbf{c} = [0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0]$.

В результате ошибки принято сообщение $\mathbf{r}_1 = [0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0]$.

Вектор, соответствующий коду $\mathbf{v} = [- \quad 1 \quad 1 \quad 1 \quad - \quad 1 \quad 1 \quad -]$.

Результат умножения $\mathbf{F} = [2 \quad -2 \quad -2 \quad -6 \quad 2 \quad 2 \quad -2 \quad 2]$.

Наименьшему значению вектора $\mathbf{F}$ соответствует координата $a = 3$, при этом F_a отрицательна, следовательно исходное сообщение формируется как $[1 : 011]$.

Тема 12. Коды Рида-Маллера

Коды Рида-Маллера – это семейство линейный помехоустойчивых кодов.

Далее будем рассматривать двоичные коды длины N как булевы (двоичные) функции от N переменных.

Таблица 25

$v_1v_2v_3$	000	001	010	011	100	101	110	111
f_1	0	1	1	0	1	1	0	0
f_2	1	0	1	0	1	0	0	1
f_1f_2	0	0	1	0	1	0	0	0

Зададим набор M всех возможных одночленов:

$M = \{1, v_1, v_2, ..., v_m, v_1v_2, ..., v_{m-1}v_m, ..., v_1v_2...v_m\}$.

Эти функции линейно независимы, как и их векторное представление.

Бинарный код Рида-Маллера R(m.r) степени r и длины 2^m содержит все линейные комбинации векторов, являющихся представлением одночленов степени не выше r от m переменных.

Для кодов первого порядка также существует ряд эффективных алгоритмов. Рассмотрим код (1,3)

Его матрица это просто все четвёрки числе от 1000 до 1111 в их бинарном представлении.

Таблица 26. Рида-Маллера R(1,3)

$v_1v_2v_3$	000	001	010	011	100	101	110	111
1	1	1	1	1	1	1	1	1
v_1	0	0	0	0	1	1	1	1
v_2	0	0	1	1	0	0	1	1
v_3	0	1	0	1	0	1	0	1

Эти векторы могут быть взяты в качестве строк образующей матрицы. Это код (8,4) d = 4, также являющийся расширенным кодом Хэмминга (с проверкой на чётности) и предназначен для исправления одинарных и обнаружения двойных ошибок.

Таблица 27. Рида-Маллера R(2,4)

v_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
v_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
v_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
v_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
v_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
v_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
v_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
v_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
v_1v_2	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
v_1v_3	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
v_1v_4	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1
v_2v_3	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
v_2v_4	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1
v_3v_4	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1

В общем случае, матрица код Рида-Маллера (r,m) содержит $k = 1 + C_m^1 + C_m^2 + ... + C_m^r$ строк. и 2^m столбцов.

Минимальное расстояние кода R(r,m) равно 2^{m-r}.

Можно выделить четыре крупные группы (частично пересекающиеся) кодов Рида-Маллера:

R(0,m) – коды повторения длины 2^m, минимальное кодовое расстояние N.

R(1,m) – помехоустойчивые коды длины 2^m, минимальное кодовое расстояние d = N/2.

R(m-1,m) – коды одной проверки на чётность длины 2^m, минимальное кодовое расстояние 2.

R(0,m) – расширенные коды Хэмминга длины 2^m, минимальное кодовое расстояние 4.

Наиболее интересной частью кодов Рида-Маллера является то, что для них существует эффективный алгоритм декодирования для любой кратности ошибок.

$$G = \begin{bmatrix} 1 & v_1 & v_2 & v_3 & v_1v_2 & v_1v_3 & v_2v_3 \end{bmatrix}^T.$$

Запишем вектор из 7 входных битов в виде

$\mathbf{m} = (m_0, m_1, m_2, m_3, m_{12}, m_{13}, m_{23})$

и вектор выходных битов

$\mathbf{c} = (c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7)$.

Операция кодирования в этом случае представляет собой умножение

$\mathbf{c} = \mathbf{mG}$.

Операция декодирования выполняется следующим образом. Для входного вектора r в первую очередь оцениваются входные биты «высшего» порядка. В данном случае это m_{12}, m_{13}, m_{23}. Затем оцениваются биты на порядок ниже и т.д.

Для оценивания используется следующий алгоритм.

$c_0 = m_0, \quad c_1 = m_0 + m_1, \quad c_2 = m_0 + m_2, \quad c_3 = m_0 + m_1 + m_2 + m_{12}.$

Сложив эти биты, получаем

$c_0 + c_1 + c_2 + c_3 = m_{12}$.

Аналогично, сложив следующие биты, получаем:

$c_4 + c_5 + c_6 + c_7 = m_{12}$.

Таким образом, с использованием полученной последовательности

$\hat{m}_{12} = r_0 + r_1 + r_2 + r_3,$

$\hat{m}_{12} = r_4 + r_5 + r_6 + r_7$.

После оценки всех битов «высшего» порядка, входной вектор пересчитывается таким образом, чтобы не учитывать эти биты:

$\mathbf{r}' = \mathbf{r} - \mathbf{m_2 G_2}$.

Затем та же процедура проводится для всех битов до самого младшего

$m_1 = c_0 + c_1,$

$m_1 = c_2 + c_3,$

$m_1 = c_4 + c_5,$

$m_1 = c_6 + c_7.$

Тема 13. Свёрточные коды. Треллис-диаграммы

Понятие свёрточных кодов

В этой лекции рассматривается важный широко используемый класс кодов, называющийся свёрточными кодами. В частности, эти коды используются стандартном 802.11 и в спутниковых коммуникациях.

Свёрточные коды работают с битами, как и все ранее рассмотренные коды, однако, в отличие от блочных кодов в систематической форме, отправитель не пересылает сообщение в виде последовательности информационных бит перемежающихся проверочными. В свёрточных кодах отправляются только проверочные биты.

Кодировщик использует скользящее окно для расчёта r>1 проверочных битов посредством комбинирования различных наборов битов внутри этого окна. Комбинирование представляет собой простое суммирование в F2 (сумма по модулю два, исключающее или), как и в предыдущих лекциях. В отличие от блочных кодов, окно перекрывает своё предыдущее положение, каждый раз сдвигаясь на 1 бит, как показано на рисунке. Размер окна в битах называется длиной кодового ограничения сверточного кода. Чем больше эта величина, тем больше число проверочных битов, на которые будет оказывать влияние каждый входной символ. Поскольку по каналу передаются только проверочные биты, большая длина кодового ограничения соответствует большей корректирующей способности кода. Взамен, процесс декодирования кодов с большой длиной кодового ограничения замедляет, что не позволяет неограниченно увеличивать эту длину.

Если свёрточный производит r проверочных битов за одно окно и перемещает окно вперёд на один бит за так, его соотношение равно 1/r. Чем больше значение r, тем выше помехоустойчивость кода, однако больше битов передаётся за такт и большая ширина канала используется для передачи. На практике, r и длину кодового

ограничения стараются выбрать как можно меньшими, обеспечивающими достаточную помехоустойчивость

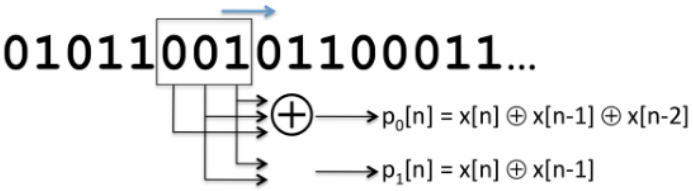

Figure 8-1: An example of a convolutional code with two parity bits per message bit ($r = 2$) and constraint length (shown in the rectangular window) $K = 3$.

Рис. 11

Будем обозначать длину кодового ограничения K.

Процесс кодирования

Кодировщик обрабатывает K битов за такт и производит r проверочных битов в соответствии с выбранной функцией, которая обрабатывает различные наборы среди этих K битов. (Предположим, что каждое сообщение содержит K-1 нулевой бит в начале для того, чтобы алгоритм функционировал корректно). Один пример приведён на рисунке и соответствует схеме с K=3 и r=2. Кодировщик производит r битов, которые затем отправляются, перемещает окно на 1 вправо и повторяет процесс.

Уравнения кодирования

Пример на рисунке демонстрирует один набор уравнений кодирования, который определяет каким образом формируются проверочные биты на основе информационных битов X. В нашем случае уравнения выглядят следующим образом:

$p_0[n]=x[n]+x[n-1]+x[n-2]$,

$p_1[n]=x[n]+x[n-1]$.

Пример уравнений кодирования для кода с r=3

$p_0[n]=x[n]+x[n-1]+x[n-2]$,

$p_1[n]=x[n]+x[n-1]$,

$p_2[n]=x[n]+ x[n-2]$.

В целом, можно заметить, что каждое уравнение кодирования представляет собой комбинацию информационных битов и образующего многочлена g. В первом примере коэффициенты образующего многочлена равны (1,1,1) и (1,1,0), а во втором (1,1,1), (1,1,0) и (1,0,1).

Обозначим g_i K-разрядный образующий многочлен для проверочного бита p_i. Можно записать p_i как

$$p_i[n] = \sum_{j=0}^{k-1} g_i[j]\, x[n-j] \bmod 2$$

Форма приведённого уравнения – свёртка g и x – отсюда возник термин «свёрточные коды». Число образующих многочленов равно числу формируемых проверочных битов r для каждого скользящего окна.

Пример.

Рассмотрим два образующих многочлена

g_0 = (1,1,1),

g_1 = (1,1,0).

В передаваемом сообщении X = [1,0,1,1,…] (предполагаем x[n] = 0 Vn<0). Тогда проверочные биты будут следующими:

$p_0[0]$=(1+0+0)=1,

$p_1[0]$=(1+0)=1,

$p_0[1]$=(0+1+0)=1,

$p_1[1]$=(0+1)=1,

$p_0[2]$=(1+0+1)=0,

$p_1[2]$=(1+0)=1,

$p_0[3]$=(1+1+0)=0,

$p_1[3]$=(1+1)=0.

И конечное сообщение, передаваемое по каналу выглядит как [1,1,1,1,0,1,0,0,…].

Существует несколько образующих многочленов, но способ их формирования за пределами этой лекции.

Несколько примеров приведено ниже.

Таблица 28. Пример образующих многочленов для r = 2

Длина ограничения	G1	G2
3	110	1110
4	1101	1110
5	11010	11101
6	110101	111011
7	110101	110101
8	110111	1110011
9	110111	111001101
10	110111001	1110011001

Два способа представления свёрточного кодера

Блочная диаграмма

Рисунок показывает тот же кодировщик, что и на рисунке в виде блочной диаграммы.

Входные биты сообщения x[n] приходят слева, «чёрный ящик» рассчитывает значения проверочных битов на основе входных битов и «состояния кодера» (предыдущих входных битов). После того, как все r выходных битов сформированы, состояние кодера сдвигается на 1, x[n] занимает место x[n-1], x[n-1] – x[n-2], и так далее, а последний бит x[n-K+1] сбрасывается.

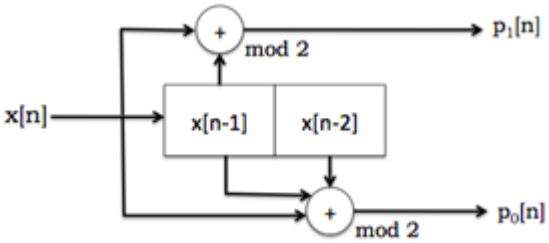

Рис. 12. Блочная диаграмма свёрточного кода

Машина конечных состояний

Машина конечных состояний *идентична* для всех кодов заданной длины K и число состояний всегда равно 2^K. Только значения выходных битов p_i зависят от конкретных количества и коэффициентов выбранных образующих многочленов.

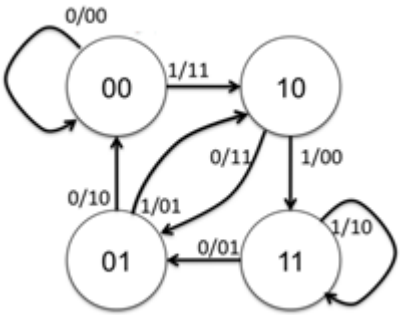

Рис. 13. Машина конечных состояний свёрточного кода

Машина конечных состояний – это способ задания процесса кодирования. Кодировщик начинает работу в исходном состоянии и обрабатывает один бит за такт. Для каждого бита сообщения состояние кодировщика меняется (или остаётся прежним) в соответствии со значением входного бита. При этом на выход подаётся определённый набор битов.

Проблема декодирования.

Проблема декодирования заключается в поиске последовательности входных битов наилучшим образом соответствующей полученному (возможно, искажённому) состоянию.

Для определения такой последовательности используется декодер, основанный на поиске максимума правдоподобия. Можно показать, что наилучшей будет такая последовательность входных информационных битов, которой соответствует наиболее близкая к полученной по расстоянию Хэмминга последовательной выходных битов.

Однако, определение такой входной последовательности в общем случае нетривиальная задача.

Например, в таблице приведены значения проверочных битов для свёрточного кода с k = 3 и r = 2. Если получатель принял 111011000110, очевидно, при передаче возникли ошибки, поскольку это сообщение не соответствует ни одной возможной

последовательности. В последнем столбце приведены значения расстояния Хэмминга для каждой из возможных передаваемых последовательностей.

Таблица 30. Пример образующих многочленов для r = 2

Входное сообщение	Выходное сообщение	Полученное сообщение	Расстояние Хэмминга
0000	000000000000	111011000110	7
0001	000000111110		8
0010	000011111000		8
0011	000011010110		4
0100	001111100000		6
0101	001111011110		5
0110	001101001000		7
0111	001100100110		6
1000	111110000000		4
1001	111110111110		5
1010	111101111000		7
1011	111101000110		2
1100	110001100000		5
1101	110001011110		4
1110	110010011000		6
1111	110010100110		3

Очевидно, что такой способ декодирования неприменим, поскольку для сообщения длины N число возможных вариантов становится равно 2^N. Для декодирования такого кода используются треллис-диаграммы.

Треллис-диаграммы

Треллис-диаграмма – это граф, узлы которого разделены на группы, представляющие собой срезы времени, а каждый узел связан с по меньшей мере одним узлом, предшествующим ему по времени, и одним узлом, следующим за ним по времени.

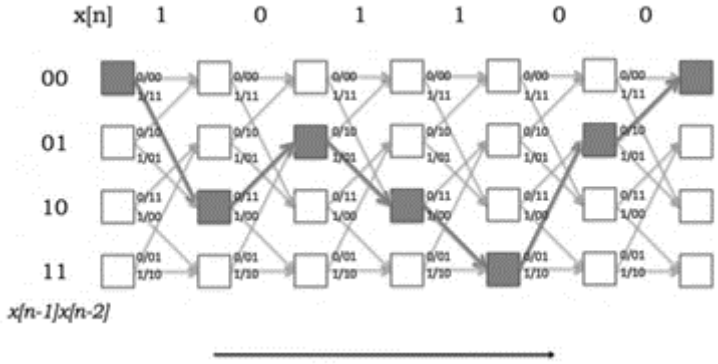

Рис. 14. Треллис-диаграмма свёрточного кода

Алгоритм Витерби

Алгоритм Витерби для свёрточных кодов заключается в минимизации метрики пути по всем возможным маршрутам на треллис-диаграмме.

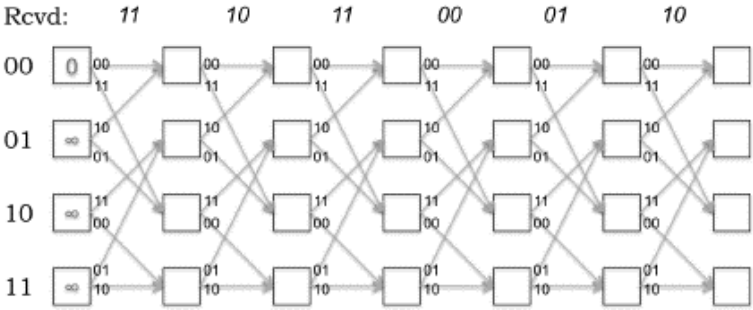

Рис. 15. Исходное состояние треллис-диаграммы

Изначально стартовому состоянию ставится в соответствие значение метрики равное нулю, остальным состояниям – бесконечно большая величина.

На каждом шаге рассчитывается вес ребра как расстояние Хэмминга между фрагментом декодируемой последовательности и выходным фрагментом соответствующего перехода в машине конечных состояний.

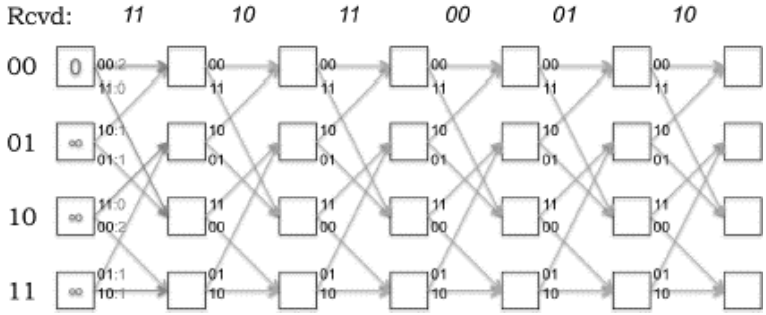

Рис. 16. Расчёт веса рёбер треллис-диаграммы

Значения метрик для каждой вершины (состояния) на следующем шаге определяется как минимум суммы веса ребра, входящего в эту вершину, и значения в вершине, из которой это ребро исходит.

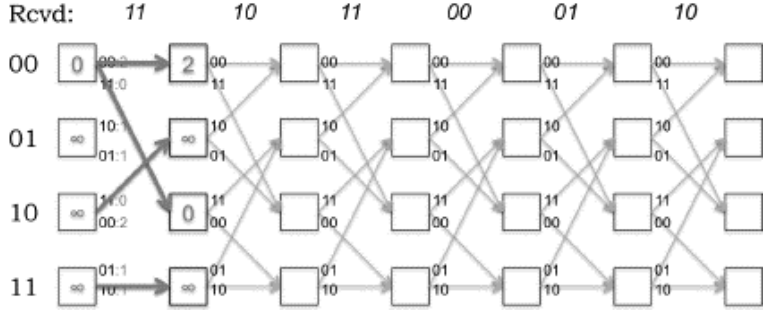

Рис. 17. Расчёт значений вершин

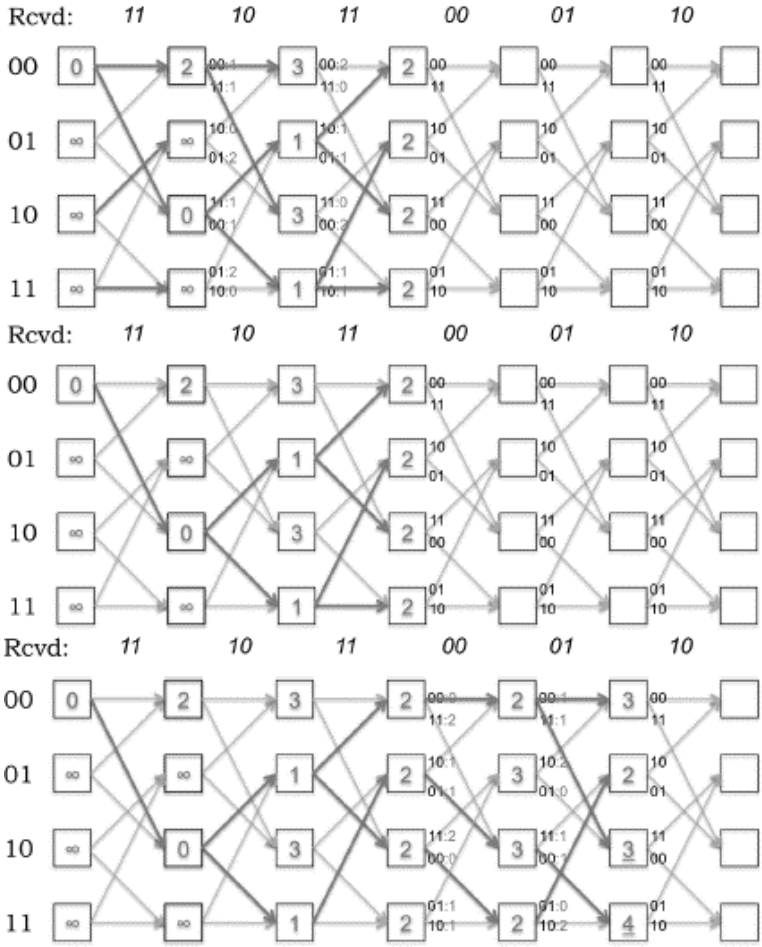

Рис. 17. Заполнение треллис-диаграммы

Таким образом заполняется вся треллис-диаграмма. Для последнего столбца состояний определяется путь, обеспечивающий получение соответствующего минимальной метрики пути до каждого состояния.

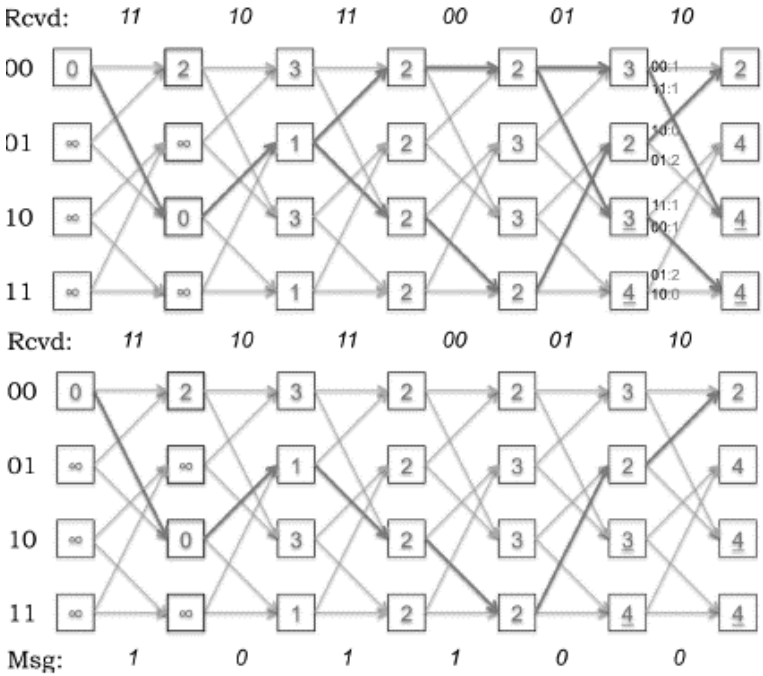

Рис. 18. Определение пути для минимального значения метрики.

Для наименьшего значения метрики пути определяются значения входных битов. Если несколько состояний содержат минимальное значение метрики, достоверное декодирование полученного сообщения невозможно.

Тема 14. Модели детерминированных сигналов

Частотное представление периодических сигналов

Рассмотрим представление детерминированных сигналов с применением в качестве базисных функций $\varphi(t) = e^{pt}$, при $p = \pm j\omega$. Такое представление называется преобразованием Фурье. В силу формулы Эйлера $\cos\omega t = \left(e^{j\omega t} + e^{-j\omega t}\right)/2$ преобразование Фурье дает возможность представить сложный сигнал в виде суммы гармоник [13].

Предположим, что функция $u(t)$, описывающая детерминированную реализацию сигнала на интервале $[t_1, t_2]$, удовлетворяет условиям Дирихле (непрерывна или имеет конечное число точек разрыва первого рода, а также конечное число экстремумов) и повторяется с периодом $T = t_2 - t_1$ при $t \in (-\infty, +\infty)$. Используя указанную выше базисную функцию $\varphi(t) = e^{\pm j\omega t}$, функцию $u(t)$ можно представить в виде

$$u(t) = \frac{1}{2}\sum_{k=-\infty}^{\infty} A(jk\omega_1)\cdot e^{jk\omega_1 t}, \qquad (45)$$

где $$A(jk\omega_1) = \frac{2}{T}\int_{t_1}^{t_2} u(t)\cdot e^{-jk\omega_1 t}dt, \qquad (46)$$

а период $T = t_2 - t_1 = 2\pi/\omega_1$.

Коэффициенты $A(jk\omega_1)$ в данном спектральном представлении называют комплексным спектром периодического сигнала $u(t)$, а значение $A(jk\omega_1)$ для конкретного k – комплексной амплитудой. Комплексный спектр дискретный, но путем замены $k\omega_1 = \omega$ для него можно построить огибающую:

$$A(j\omega) = \frac{2}{T}\int_{t_1}^{t_2} u(t)\cdot e^{-j\omega t}dt. \qquad (47)$$

Как всякое комплексное число, комплексный спектр можно представить:

а) в показательной форме:

$$A(jk\omega_1) = A(k\omega_1)\cdot e^{-j\varphi(k\omega_1)}, \qquad (48)$$

где $A(k\omega_1)$ – спектр амплитуд, а $\varphi(k\omega_1)$ – спектр фаз (также дискретный);

б) в алгебраической форме:

$$A(jk\omega_1) = A_k - jB_k, \qquad (49)$$

где

$$A_k = \frac{2}{T}\int_{t_1}^{t_2} u(t)\cdot\cos(k\omega_1 t)\,dt,\ B_k = \frac{2}{T}\int_{t_1}^{t_2} u(t)\cdot\sin(k\omega_1 t)\,dt.$$

Представление (49) получается из (46) путем замены по формуле Эйлера: $e^{-jk\omega_1 t} = \cos(k\omega_1 t) - j\sin(k\omega_1 t)$. Ясно, что $A(k\omega_1) = \sqrt{A_k^2 + B_k^2}$, а $\varphi(k\omega_1) = arctg(B_k / A_k)$. Из равенства, определяющего в (10.12) вещественную часть A_k при $k = 0$, получаем равенство для постоянной составляющей сигнала:

$$\frac{A_0}{2} = \frac{1}{T}\int_{t_1}^{t_2} u(t)\,dt. \qquad (50)$$

Объединяя в (45) комплексно-сопряженные составляющие можно получить ряд Фурье в тригонометрической форме:

$$u(t) = \frac{A_0}{2} + \frac{1}{2}\sum_{k=1}^{\infty}\left[A(jk\omega_1)\cdot e^{jk\omega_1 t} + A(-jk\omega_1)\cdot e^{-jk\omega_1 t}\right] =$$

$$= \frac{A_0}{2} + \frac{1}{2}\sum_{k=1}^{\infty}\left[A(k\omega_1)\cdot e^{j\left[k\omega_1 t - \varphi(k\omega_1)\right]} + A(k\omega_1)\cdot e^{-j\left[k\omega_1 t - \varphi(k\omega_1)\right]}\right] =$$

$$= \frac{A_0}{2} + \sum_{k=1}^{\infty} A(k\omega_1)\cos(k\omega_1 t - \varphi(k\omega_1)). \qquad (51)$$

Спектры амплитуд – $A(k\omega_1)$ и фаз – $\varphi(k\omega_1)$ могут быть представлены спектральными диаграммами в виде совокупности

линий, каждая из которых соответствует определенной частоте (одному из слагаемых). Поэтому эти спектры называют линейчатыми. Сигналы, линейчатые спектры которых включают гармоники некратных частот, называются почти периодическими.

Частотное представление непериодических сигналов

Предположим, что соответствующая реальному непериодическому сигналу функция $u(t)$ удовлетворяет условиям Дирихле и абсолютно интегрируема: $\int\limits_{-\infty}^{\infty} |u(t)| \cdot dt < \infty$. Тогда спектральное представление непериодического сигнала $u(t)$ можно строить путем увеличения периода периодического сигнала до бесконечности. Для этого поступим следующим образом.

Подставим выражение (46) для комплексной амплитуды $A(jk\omega_1)$ периодического сигнала в (45). С учетом того, что $T = 2\pi / \omega_1$ имеем

$$u(t) = \frac{1}{2} \sum_{k=-\infty}^{\infty} \left[\frac{\omega_1}{\pi} \int\limits_{t_1}^{t_2} u(t) \cdot e^{-jk\omega_1 t} dt \right] \cdot e^{jk\omega_1 t} . \qquad (52)$$

Далее осуществим предельный переход при $T \to \infty$. При этом сумма переходит в интеграл, $\omega_1 = \Delta\omega \to d\omega$, $k\omega_1 \to \omega$. В результате получаем:

$$u(t) = \frac{1}{2\pi} \int\limits_{-\infty}^{\infty} \left[\int\limits_{-\infty}^{\infty} u(t) \cdot e^{-j\omega t} dt \right] \cdot e^{j\omega t} d\omega .$$

Введя в последнем равенстве для интеграла в квадратных скобках обозначение $S(j\omega)$, запишем пару преобразований Фурье:

$$u(t) = \frac{1}{2\pi} \int\limits_{-\infty}^{\infty} S(j\omega) \cdot e^{j\omega t} d\omega , \qquad (53)$$

$$S(j\omega) = \int\limits_{-\infty}^{\infty} u(t) \cdot e^{-j\omega t} dt . \qquad (54)$$

Комплексную функцию $S(j\omega)$ называют комплексной спектральной плотностью или спектральной характеристикой. Также

как в случае периодического сигнала, для непериодического сигнала имеют место следующие представления спектральной характеристики:

а) показательная форма:

$$S(j\omega) = S(\omega) \cdot e^{-j\varphi(\omega)}, \qquad (55)$$

где $S(\omega) = |S(j\omega)|$ – спектральная плотность амплитуд, а $\varphi(\omega)$ – спектр фаз;

б) алгебраическая форма (получается из (54) путем замены $e^{-j\omega t} = \cos(\omega t) - j\sin(\omega t)$):

$$S(j\omega) = A(\omega) - jB(\omega), \qquad (56)$$

где

$$A(\omega) = \int_{-\infty}^{+\infty} u(t) \cdot \cos(\omega t)\, dt, \; B(\omega) = \int_{-\infty}^{+\infty} u(t) \cdot \sin(\omega t)\, dt. \qquad (57)$$

При этом

$$S(\omega) = |S(j\omega)| = \sqrt{|A(\omega)|^2 + |B(\omega)|^2}, \quad \varphi(\omega) = arctg\left[B(\omega)/A(\omega)\right]. \qquad (58)$$

Подставляя $S(j\omega)$ из (10.18) в (10.16) имеем

$$u(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) \cdot e^{j\left[\omega t - \varphi(\omega)\right]} d\omega =$$

$$\frac{1}{2\pi}\left[\int_{-\infty}^{\infty} S(\omega) \cdot \cos\left[\omega t - \varphi(\omega)\right] d\omega + j\int_{-\infty}^{\infty} S(\omega) \cdot \sin\left[\omega t - \varphi(\omega)\right] d\omega\right].$$

Второй интеграл от нечетной функции равен нулю, а первый (в силу четности подынтегральной функции) можно записать только для положительных частот. Таким образом, получаем тригонометрическую форму ряда Фурье:

$$u(t) = \frac{1}{\pi}\int_{0}^{\infty} S(\omega) \cdot \cos\left[\omega t - \varphi(\omega)\right] d\omega, \qquad (59)$$

которая дает возможность ясного физического толкования.

В заключение рассмотрим еще одно интересное свойство. Для функции $u(t)$, заданной на интервале $[t_1, t_2]$ в соответствии с (10.17) можно записать

$$S(j\omega) = \int_{t_1}^{t_2} u(t) \cdot e^{-j\omega t} dt . \qquad (60)$$

Сравнивая правые части (47) и (60) нетрудно заметить, что имеет место равенство $A(j\omega) = \frac{2}{T} \cdot S(j\omega)$, т.е. по $S(j\omega)$ одиночного импульса можно построить линейчатый спектр их периодической последовательности.

Соотношение между длительностью сигналов и шириной их спектров

Предположим, что сигнал $u(t)$ определенной продолжительности имеет спектральную характеристику $S(j\omega)$. Найдем соответствующую характеристику $S_\lambda(j\omega)$ для сигнала $u(\lambda t)$, длительность которого изменена в λ раз:

$$S_\lambda(j\omega) = \int_{-\infty}^{\infty} u(\lambda t) \cdot e^{-j\omega t} dt = \frac{1}{\lambda} \int_{-\infty}^{\infty} u(\tau) \cdot e^{-j\frac{\omega\tau}{\lambda}} d\tau = \frac{1}{\lambda} S\left(j\frac{\omega}{\lambda}\right), \qquad (61)$$

где $\tau = \lambda t$.

Из (61) видно, что спектр укороченного (удлиненного) в λ раз сигнала в λ раз шире (уже), при этом коэффициент $1/\lambda$ изменяет только амплитуды гармоник и на ширину спектра не влияет. Указанное свойство связано с тем, что переменные t и ω входят в показатель степени экспоненциальной функции прямого и обратного преобразования Фурье в виде произведения. Из этого следует, что длительность сигнала и ширина его спектра не могут быть одновременно ограничены конечными интервалами. В частности, имеет место соотношение:

$\Delta t \cdot \Delta f = Const$,

где Δt – длительность импульса, Δf – ширина спектра.

Тема 15. Восстановление сигнала по его дискретным значениям

Формулировка задач дискретизации и восстановления

Дискретизация сигнала – это преобразование функции непрерывного аргумента в функцию дискретного времени. Она заключается в замене непрерывного сигнала $u(t)$ совокупностью координат:

$$\left[c_1, c_2, ..., c_N\right] = A\left[u(t)\right], \tag{62}$$

где $A[\cdot]$ – некоторый оператор.

С точки зрения простоты реализации целесообразно использовать линейные операторы. В частности, для определения координат сигнала удобно использовать соотношение

$$c_i = Au(t) = \int_T \varphi_i(t) \cdot u(t) \cdot dt, \quad i = \overline{1, N}, \tag{63}$$

где $\varphi_i(t), \quad i = \overline{1, N}$ – заданные базисные (в частности, могут использоваться ортогональные) функции.

Дискретизация по соотношению (63), вследствие применения операции интегрирования, обладает высокой помехоустойчивостью. Однако при этом имеет место задержка сигнала на время интегрирования T. Поэтому чаще дискретизация сводится к замене сигнала совокупностью его мгновенных значений (выборок):

$$c_i = u(t_i). \tag{64}$$

Это достигается использованием в (63) дельта-функции: $\varphi_i(t) = \delta(t - t_i)$.

Представление непрерывного сигнала совокупностью равноотстоящих отсчетов в виде решетчатой функции:

$$u_g(t) = \sum_{k=-\infty}^{\infty} u(t) \cdot \delta(t - k\Delta t) \tag{65}$$

– наиболее распространенный вид дискретизации. Функция $u_g(t)$ равна $u(k\Delta t)$ в точках $t = k\Delta t$ и нулю в остальных точках. Если шаг дискретизации $\Delta t_i = t_i - t_{i-1} = Const$ – дискретизация называется равномерной. Пределы суммирования в (65) могут быть установлены конечными, исходя из условий физической реализуемости.

Обычно дискретизация осуществляется с целью дальнейшего преобразования сигнала в цифровую форму. В результате цифрового кодирования дискретного сигнала происходит его квантование – замена в соответствующие моменты времени мгновенных значений сигнала ближайшими разрешенными. При этом сигнал оказывается дискретным как по времени, так и по множеству значений.

Важное достоинство цифровой формы представления сигнала состоит в том, что много уровней квантования можно представить небольшим количеством разрядов. Кроме того, при представлении в цифровой форме могут быть реализованы сложные алгоритмы обработки на ЭВМ, включая построение кодов обнаруживающих и исправляющих ошибки.

При последующем использовании дискретного сигнала для целей управления обычно осуществляют его восстановление с использованием некоторого заданного оператора:

$$u^*(t) = B[c_1, c_2, ..., c_N], \qquad (66)$$

Если дискретизация осуществлялась оператором вида (63) с использованием ортогональных функций $\varphi_i(t)$, $i = \overline{1, N}$, для восстановления непрерывного сигнала может использоваться оператор

$$u^*(t) = \sum_{i=1}^{N} c_i \varphi_i(t). \qquad (67)$$

Для оценки качества восстановления сигнала используются следующие критерии.

Равномерное приближение (критерий наибольшего отклонения):

$$\max_{t \in T} \left| u(t) - u^*(t) \right| \le \varepsilon_{доп}.$$

Равномерное приближение для ансамбля реализаций:

$$\sup_{u_i(t)\in U}\left|u_i(t)-u_i^*(t)\right|\le\varepsilon_{доп}.$$

Критерий среднеквадратического отклонения (СКО):

$$\sigma=\sqrt{\frac{1}{T}\int_T\left|u(t)-u^*(t)\right|^2dt}\le\sigma_{доп}$$.СКО для ансамбля N реализаций – σ_Σ вычисляется усреднением по ансамблю с учетом вероятностей реализаций p_i, $i=\overline{1,N}$:

$$\sigma_\Sigma=\sum_{i=1}^{N}p_i\sigma_i\le\sigma_{\Sigma,доп}.$$

Интегральный критерий:

$$\varepsilon=\frac{1}{T}\int_T\left|u(t)-u^*(t)\right|dt\le\varepsilon_{доп}.$$

Величину интегрального критерия ε_Σ для N реализаций вычисляют путем усреднения по ансамблю:

$$\varepsilon_\Sigma=\sum_{i=1}^{N}p_i\varepsilon_i.$$

Применяют также вероятностный критерий, определяемый как допустимый уровень вероятности $\mathrm{P}_{доп}$ того, что ошибка не превысит допустимого значения $\varepsilon_{доп}$:

$$\mathrm{P}\left\{\left|u(t)-u^*(t)\right|\le\varepsilon_{доп}\right\}\le\mathrm{P}_{доп}.$$

Использование одного из указанных критериев в каждом конкретном случае зависит от требований к системе и доступной априорной информации.

Теорема Котельникова

Как отмечалось выше, наиболее широко используется равномерная дискретизация. При этом для выбора величины шага дискретизации используется модель сигнала в виде эргодического случайного процесса, каждая реализация которого представляет

собой функцию с ограниченным спектром. Теоретической основой этого подхода является следующая теорема Котельникова.

Любая функция $u(t)$, допускающая преобразование Фурье и имеющая непрерывный спектр, ограниченный полосой частот от 0 до $f_c = \omega_c / 2\pi$, полностью определяется дискретным рядом своих мгновенных значений, отсчитанных через интервалы времени $\Delta t = 1/(2 \cdot f_c) = \pi / \omega_c$.

Доказательство. Поскольку по предположению функция $u(t)$ имеет ограниченный спектр, т.е. $S(j\omega) = 0$ при $|\omega| > \omega_c$, можно записать равенство

$$u(t) = \frac{1}{2\pi} \int_{-\omega_c}^{+\omega_c} S(j\omega) \cdot e^{j\omega t} d\omega. \qquad (68)$$

Функцию $S(j\omega)$ на *конечном интервале* $[-\omega_c, \omega_c]$ можно разложить в ряд Фурье. Пару преобразований Фурье запишем, полагая $S(j\omega)$ условно продолжающейся с периодом $2\omega_c$ и формально заменив в (45), (46) t на ω, а ω_1 на $\Delta t = \pi / \omega_c$:

$$S(j\omega) = \frac{1}{2} \sum_{-\infty}^{+\infty} A_k \cdot e^{jk\Delta t\omega}, \qquad (69)$$

$$A_k = \frac{1}{\omega_c} \int_{-\omega_c}^{\omega_c} S(j\omega) \cdot e^{-jk\Delta t\omega} d\omega. \qquad (70)$$

Сравним соотношения (70) и (68), предварительно переписав равенство (68) для дискретных моментов времени $t_k = k\Delta t$:

$$u(k\Delta t) = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} S(j\omega) \cdot e^{j\omega k\Delta t} d\omega. \qquad (71)$$

Нетрудно заметить, что

$$A_k = \frac{2\pi}{\omega_c} \cdot u(-k\Delta t). \qquad (72)$$

Подставляя значение A_k из (72) в (69) можно записать:

$$S(j\omega)=\frac{\pi}{\omega_c}\sum_{-\infty}^{+\infty}u(-k\Delta t)\cdot e^{jk\Delta t\omega}.$$

В последнем равенстве знак минус перед k можно поменять на обратный, т.к. суммирование ведется как по положительным, так и по отрицательным числам:

$$S(j\omega)=\frac{\pi}{\omega_c}\sum_{-\infty}^{+\infty}u(k\Delta t)\cdot e^{-jk\Delta t\omega}. \qquad (73)$$

Теперь подставим $S(j\omega)$ из (73) в (68):

$$u(t)=\frac{1}{2\omega_c}\int_{-\omega_c}^{+\omega_c}\left(\sum_{-\infty}^{+\infty}u(k\Delta t)\cdot e^{-jk\Delta t\omega}\right)\cdot e^{j\omega t}d\omega=\frac{1}{2\omega_c}\sum_{-\infty}^{+\infty}u(k\Delta t)\int_{-\omega_c}^{+\omega_c}e^{j\omega(t-k\Delta t)}d\omega.$$

После выполнения интегрирования в правой части последнего равенства получаем

$$u(t)=\sum_{-\infty}^{+\infty}u(k\Delta t)\frac{\sin\omega_c(t-k\Delta t)}{\omega_c(t-k\Delta t)}=\sum_{-\infty}^{+\infty}u(k\Delta t)\operatorname{sinc}\omega_c(t-k\Delta t). \qquad (74)$$

Итак, мы выразили функцию $u(t)$ через ее дискретные значения, взятые в моменты времени $t_k=k\Delta t$. Предположим $t=n\Delta t$, где n – некоторое целое число. Поскольку $\Delta t=\pi/\omega_c$, для любых целых k и n

$$\omega_c(n\Delta t-k\Delta t)=(n-k)\omega_c\Delta t=(n-k)\pi.$$

Следовательно

$$\frac{\sin\omega_c(t-k\Delta t)}{\omega_c(t-k\Delta t)}=\begin{cases}1, & \text{если} \quad t=k\Delta t,\\ 0, & \text{если} \quad t=n\Delta t, \quad n\neq k.\end{cases}$$

Это означает, что значения функции $u(t)$ в моменты времени $t_k=k\Delta t$ представляют собой не что иное, как ее отсчеты. Таким образом, функция с ограниченным спектром может быть представлена рядом (74), коэффициенты которого представляют собой отсчеты значений функции, взятые через интервалы времени

$$\Delta t=\frac{\pi}{\omega_c}=\frac{1}{2\cdot f_c}. \qquad (75)$$

На основании этого можно представить следующую схему передачи-приема. На передающей стороне мгновенные значения сигнала $u(t)$ передаются через интервалы Δt, определяемые по соотношению (75). На приемной стороне последовательность импульсов пропускают через идеальный фильтр нижних частот с частотой среза f_c. Тогда при длительной передаче теоретически сигнал на выходе фильтра будет точно воспроизводить переданный непрерывный сигнал $u(t)$.

В действительности реальный сигнал всегда имеет конечную длительность, следовательно, его спектр неограничен. Ошибка возникает не только за счет принудительного ограничения спектра, но и за счет конечного числа отсчетов в интервале времени T, которых в соответствии с теоремой будет $N = 2 f_c T$.

Модель сигнала с ограниченным спектром имеет также принципиальное теоретическое неудобство. Она не может отражать основное свойство сигнала – способность нести информацию. Дело в том, что поведение функции с ограниченным спектром можно точно предсказать на всей оси времени, если она точно известна на сколь угодно малом отрезке времени.

Тем не менее, теорема Котельникова имеет важное прикладное значение. На практике ширину спектра f_c определяют как интервал частот, вне которого спектральная плотность меньше некоторой заданной величины. При таком допущении функция на интервале T с некоторой степенью точности (зависящей от точности представления спектральной плотности) определяется посредством $N = 2 f_c T$ отсчетов, т.е. общий смысл теоремы Котельникова сохраняется.

Квантование сигналов

Физически реализуемый непрерывный сигнал $u(t)$ всегда ограничен некоторым диапазоном $[u_{\min}, u_{\max}]$. Вдобавок часто устройство может воспроизводить лишь конечное множество фиксированных значений сигнала из этого диапазона. В частности,

непрерывная шкала мгновенных значений $u_n = u_{\max} - u_{\min}$ может быть разбита на n одинаковых интервалов, а разрешенные значения сигнала равноотстоят друг от друга, тогда говорят о равномерном квантовании. Если постоянство интервала (шага квантования) не соблюдается, то квантование неравномерное.

Из множества мгновенных значений, принадлежащих i-му интервалу (шагу квантования), только одно значение u_i' является разрешенным (i-й уровень квантования), а любое другое округляется до u_i'. Предположим, равномерное квантование с шагом $\Delta = (u_{\max} - u_{\min})/n$ осуществляется так, что уровни квантования u_i' размещаются в середине каждого шага. Ясно, что при этом ошибка квантования минимальна и не превышает $0{,}5\Delta$. Определим для этого случая среднеквадратическое отклонение (СКО) ошибки квантования.

В общем случае СКО ошибки квантования σ_i для i-го шага определяется соотношением

$$\sigma_i = \sqrt{\int_{u_{i-1}}^{u_i} \left(u(t) - u_i'\right)^2 w(u)\,du}\,, \qquad (76)$$

где $w(u)$ – функция плотности вероятности мгновенных значений сигнала U. Если шаги квантования малы по сравнению с диапазоном изменения сигнала, плотность $w(u)$ в пределах каждого шага можно считать постоянной и равной, например, $w(u_i')$. Тогда, вводя новую переменную $y = u(t) - u_i'$, для указанного способа квантования в соответствии с (76) имеем

$$\sigma_i = \sqrt{w(u_i') \int_{-\frac{\Delta_i}{2}}^{\frac{\Delta_i}{2}} y_i^2\,dy_i} = \sqrt{w(u_i')\frac{\Delta_i^3}{12}}\,. \qquad (77)$$

С учетом того, что $p\left(u_i^{'}\right)>0$ и $\Delta_i>0$ для всех $i=\overline{1,n}$ в соответствии с (77) можно записать дисперсию ошибки квантования на i-м шаге:

$$\sigma_i^2=\left[w\left(u_i^{'}\right)\Delta_i\right]\frac{\Delta_i^2}{12}. \qquad (78)$$

Оказывается, она равна величине $\Delta_i^2/12$, умноженной на вероятность $w\left(u_i^{'}\right)\Delta_i$ попадания мгновенного значения сигнала в данный интервал. Дисперсия полной ошибки определяется как математическое ожидание дисперсий $\Delta_i^2/12$ на отдельных шагах:

$$\sigma^2=\sum_{i=1}^{n}\left[w\left(u_i^{'}\right)\Delta_i\right]\frac{\Delta_i^2}{12}.$$

Если интервалы одинаковы, т.е. $\Delta_i=\Delta$ для всех $i=\overline{1,n}$, с учетом условия нормировки $\sum_{i=1}^{n}\left[w\left(u_i^{'}\right)\Delta\right]=1$, получаем

$$\sigma^2=\frac{\Delta^2}{12}\sum_{i=1}^{n}\left[w\left(u_i^{'}\right)\Delta\right]=\frac{\Delta^2}{12}.$$

Если на квантуемый сигнал воздействует помеха, он может попасть в интервал, соответствующий другому уровню квантования. Интуитивно ясно (и это можно строго показать), что в случае, когда помеха ξ имеет равномерное распределение $w(\xi)=1/a$, где $a/2$ – амплитуда помехи, симметричной относительно мгновенного значения сигнала, вероятность неправильного квантования сигнала резко возрастает при $a>\Delta$. Воздействие нормально распределенной помехи с параметрами $\left(0,\sigma^2\right)$ эквивалентно воздействию равномерно распределенной помехи при $a=3\sigma$.

Список источников

1. *Фурсов, В.А.* Теория информации [Текст]: учеб. пособие / В.А. Фурсов. – Самара: Изд-во СГАУ, 2013. – 128 с.
2. *Arndt, C.* Information Measures, Information and its Description in Science and Engineering [Текст] / C. Arndt. – Springer Series: Signals and Communication Technology, 2004. – 603 p.
3. *Cover, T.* Elements of information theory (2-nd ed.) [Текст] / T. Cover, J.A. Thomas. – New York: Wiley-Interscience, 2006. – 776 p.
4. *MacKay, D.J.C.* Information Theory, Inference, and Learning Algorithms [Текст] / MacKay D.J.C. – Cambridge: Cambridge University Press, 2003. – 640 p.
5. *McEliece, R.* The Theory of Information and Coding [Текст] / R. McElliece. – Cambridge, 2002. – 410 p.
6. *Yeung, R.W.* A First Course in Information [Текст] / R.W. Yeung. – Theory Kluwer Academic/Plenum Publishers, 2002. – 431 p.
7. *Скляр, Б.* Цифровая связь. Теоретические основы и практическое применение [Текст] / Б. Скляр. – 2-е изд., испр.; пер. с англ. – М.: Издательский дом “Вильямс”, 2003. – 1104 с.
8. *Шеннон, К.* Работы по теории информации и кибернетике [Текст] / К. Шеннон. – М. : Изд-во иностранной литературы, 1963. – 830 с.
9. *Дмитриев В.И.* Прикладная теория информации [Текст]: учеб. пособие. – М.: Высшая школа, 1989.– 320 с.
10. Университет ИТМО. Алгоритм LZW [Электронный ресурс]. – Режим доступа: http://neerc.ifmo.ru/ – Заглавие с экрана. – (Дата обращения: 04.09.2018).

Учебное издание

Гошин Егор Вячеславович

ТЕОРИЯ ИНФОРМАЦИИ И КОДИРОВАНИЯ

Учебное пособие

В авторской редакции
Компьютерная вёрстка А.В. Ярославцевой

Подписано в печать 27.09.2018. Формат 60×84 1/16.
Бумага офсетная. Печ. л. 7,75.
Тираж 100 экз. Заказ .

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)
443086, САМАРА, МОСКОВСКОЕ ШОССЕ, 34.

Изд-во Самарского университета.
443086 Самара, Московское шоссе, 34.