НАУЧНЫЙ ЖУРНАЛ
НАУКА И МИРОВОЗЗРЕНИЕ

ТЕОРИЯ КОДИРОВАНИЯ. ВИДЫ КОДИРОВАНИЯ. ОПТИМАЛЬНЫЕ КОДЫ

Ширлиев Мухамметсадык

Преподаватель, Международного университета нефти и газа имени Ягшыгелди Какаева, г. Ашхабад Туркменистан

Худайбердиева Умыда

Студент, Международного университета нефти и газа имени Ягшыгелди Какаева, г. Ашхабад Туркменистан

Акадова Чынар

Студент, Международного университета нефти и газа имени Ягшыгелди Какаева, г. Ашхабад Туркменистан

Ханова Айнагозел

Студент, Международного университета нефти и газа имени Ягшыгелди Какаева, г. Ашхабад Туркменистан

Теория кодирования

На прошлой лекции мы ввели определение средней длины кода. Вспомнем это определение.

Определение (средняя длина кода) Для некоторого источника, первичного алфавита A, состоящего из N символов, вторичного алфавита B и кода j определяется как взвешанная сумма кодов для всех символов первичного алфавита, взвешанных соответствующими вероятностями появления этих символов в сообщении.

$K(\varphi,A,B) = \sum^{N}{}_{i=1} n_i p_i,$

Где n_i – это длина кодового слова для i-го символа первичного алфавита.

Мы отметили, что Так как

nI^A £ mI^B,

минимальное возможное значение длины кода будет

$K_{min}(A,B) = I^A / I^B$

В качестве меры превышения длины кода мы будем использовать относительную избыточность кода.

Определение (Относительная избыточность кода) Относительной избыточностью кода мы будем называть величину

$Q(\varphi,A,B) = (K(\varphi,A,B) - K_{min}(A,B))/ K_{min}(A,B) = K(\varphi,A,B) / K_{min}(A,B) – 1 = K(\varphi,A,B)\ I^B/ I^A – 1$

Для фиксированного первичного алфавита A и фиксированного вторичного алфавита B существует множество различных способов построения кода, ставящего символам из алфавита A символы или комбинации символов алфавита B. Для разных вариантов относительная избыточность тоже может быть различной. Кроме того ресурсы (машинная память, время работы), необходимые для кодирования/декодирования разных кодов могут быть различны. Однако, мы не будем рассматривать алгоритмическую оптимальность в рамках Теории Кодирования.

Определение (Оптимальный код) Для некоторого первичного алфавита A и вторичного алфавита B асимптотически оптимальным кодом будем называть такой способ кодирования, при котором, если длина сообщений стремится к бесконечности, избыточность кодирования стремится к нулю.

Насколько хороший код можно построить? На этот вопрос дает ответ первая теорема Шеннона:

Теорема (Основная теорема кодирования при отсутствии шумов)

При отсутствии шумов всегда возможен такой вариант кодирования сообщения, при котором относительная избыточность кода будет сколь угодно близка к нулю.

Первая теорема Шеннона дает нам уверенность в возможности оптимального кодирования. Но не дает рецепта построения такого кода. Именно поэтому теория кодирования не становится тривиальной даже в случае отсутствия шумов.

Кроме уменьшения избыточности кодирования для заданных первичного и вторичного алфавита, уменьшить длину кода можно и правильным выбором

Как видно из определения, существует всего два способа уменьшить минимально возможное значение средней длины кода $K_{min}(A,B) = I^A / I^B$: уменьшить числитель и увеличить знаменатель. Как это сделать? Вспомним, что сообщения, записанные символами первичного алфавита генерируются некоторым источником S, который характеризуется вероятностями появления отдельных символов алфавита A на выходе источника. Учитывая особенности источника можно достичь нашей цели:

• Уменьшить числитель. Этого можно достичь, если учесть различие символов первичного алфавита, корреляции нескольких знаков.

• Увеличить знаменатель. Этого можно достичь, если использовать такой способ кодирования, при котором появление знаков вторичного алфавита было бы равновероятным, то есть $I^B=\log_2|B|$.

Давайте учитывать различные вероятности появления отличных символов первичного алфавита на выходе источника. Но мы будем считать, что корреляций между символами, генерируемыми источником нет. То есть, источник не запоминает, какие символы он уже выдавал, и генерирует новые символы независимо от прошлого. Такие источники называют источниками без памяти. Тогда, минимальное значение средней длины кода можно записать следующим образом:

$K_{min}(A,B) = I^A / \log_2|B|$.

На практики почти повсеместно в цифровой технике используется двоичное кодирование, то есть $|B|=2$, а сам вторичный алфавит B состоит из нуля и единицы $B=\{0,1\}$. Такое кодирование проще всего реализовать.

Например, информацию можно хранить как последовательность намагниченных или немагниченных участков жесткого диска. Нетрудно видеть, что в этом случае

$K_{min}(A,2) = I^A$

Для двоичного кодирования первая теорема Шеннона может быть переформулирована.

Теорема (Вариант теоремы Шеннона). При отсутствии помех средняя длина двоичного кода может быть сколь угодно близкой к средней информации, приходящейся на один символ первичного алфавита.

Формула относительной избыточности кодирования в случае двоичного кода принимает вид:

$Q(\varphi,A,2) = K(\varphi,A,2) / I^A - 1$

Классификация способов кодирования

При кодировании можно использовать следующие особенности вторичного алфавита:

• Элементарные сигналы (0, 1) могут иметь одинаковые длительности, или разные длительности.

• Длина кода может быть одинаковой для всех знаков первичного алфавита (равномерный код), или быть различной (неравномерный код).

• Можно кодировать каждый знак первичного алфавита (алфавитное кодирование), или кодировать блоки символов первичного алфавита (блочное кодирование).

Комбинации перечисленных способов определяют основу конкретного способа кодирования. Мы познакомимся с несколькими такими методами.

Неравномерный код с разделителем

Будем кодировать каждую букву первичного алфавита отдельно. Условимся, что разделителем отдельных кодов будет последовательность двух нулей “00” (признак конца знака). А разделителем слов будет последовательность трех нулей “000” (признак конца слова, пробел). Я предлагаю вам следующие правила построения кодов:

• Код признака конца буквы можно включить в код буквы, так как код признака конца буквы не используется сам по себе, отдельно от кода буквы. То есть все коды букв будут заканчиваться на «00».

• Коды букв не должны содержать двух (или более) нулей подряд нигде кроме как в конце кода. Иначе такой код будет воспринят как два кода двух различных знаков первичного алфавита.

• Код буквы, кроме Иначе ноль в начале кода буквы и два нуля в конце кода буквы слева образуют код пробела помимо нашего желания!

• Разделителю слов «000» всегда предшествует признак конца знака. То есть ы конце каждого слова реализуется последовательность «00000». Значит коды букв могут оканчиваться не только на «00», но и на «000», и на «0000». При этом не будет возникать трудностей с правильным декодированием сообщения.

В соответствии с перечисленными правилами можно построить таблицу кодов для кириллицы.

код	Символ	P(i)	код	Символ	P(i)	код	Символ	P(i)
000	_	0.175	110100	Л	0.035	1111000	Б	0.014
100	О	0.090	111000	К	0.028	1111100	Г	0.012
1000	Е	0.072	111100	М	0.026	10101000	Ч	0.012
1100	Ё	0.072	1010000	Д	0.025	10101100	Й	0.010
10000	А	0.062	1010100	П	0.023	10110000	Х	0.009

У нас есть формула, по которой мы можем найти среднюю длину кода для данного способа кодирования:

$K(Code1,Cyr,2)= \sum^{34}_{I=1} P_i k_i = 5,5$

где k_i - это длина i-го кодового слова.

Таким образом избыточность кодирования в данном случае равна:

$Q(Code1,Cyr,2) = K(Code1,Cyr,2) / I^A – 1 = 5,5 / 4,72 - 1 \sim 0,17$

Это значит, что при данном способе кодирования мы будем вынуждены передавать на 17% больше информации, чем содержится в исходном сообщении. В итоге загружая канал связи больше, чем требуется!

Существует ли способ кодирования при котором отпала бы необходимость в разделителе знаков?

Префиксные коды

Оказывается, такое возможно!

Условие Фано. Неравномерный код может быть однозначно декодирован, если никакой из кодов не совпадает с началом (префиксом) какого-либо другого, более длинного кода.

Например: Если имеется код «110» то в качестве кодов нельзя использовать последовательности «1», «11», но можно использовать «0», «10» и пр.

Рассмотрим пример префиксного кода:

а	л	м	р	у	ы
10	010	00	11	0110	0111

Попробуйте декодировать сообщение: 00100010000111010101110000110

Начинать следует слева направо, последовательно вычеркивая обнаруженные коды, и записывая соответствующие им знаки первичного алфавита.

Префиксный код Шеннона-Фано

В 1948-1949 гг. Клод Шеннон и Роберт Фано независимо предложили префиксный код, названный в последствие в их честь.

Рассмотрим этот префиксный код на примере. Пусть имеется первичный алфавит: a_1, a_2, … a_6 с вероятностями появления этих символов в сообщении соответственно 0,3; 0,2; 0,2; 0,15; 0,1; 0,05. Расположим эти знаки в таблице в порядке убывания их вероятностей.

Знак	Вероятность (p_i)	Разряды кода				Код
		1	2	3	4	
a_1	0,30	0	0			00
a_2	0,20	0	1			01
a_3	0,20	1	0			10
a_4	0,15	1	1	0		110
a_5	0,10	1	1	1	0	1110
a_6	0,05	1	1	1	1	1111

Кодирование осуществляется следующим образом. Все знаки делятся на две группы, так чтобы суммы вероятностей в каждой группе были приблизительно равны. В нашем примере в первую группу попадают знаки a1, a2, все остальные знаки попадают в другую группу. Установим в ноль первый знак кодов для всех символов из первой группы, и установим равным единицы первый знак кодов всех символов из второй группы. Продолжим деление каждой группы по той же схеме до тех пор, пока не получим группы, состоящие из одного элемента. Эта процедура изображена в таблице.

Полученный код удовлетворяет условию Фано, следовательно он является префиксным.

Средняя длина этого кода равна

К(Шеннона-Фано, А,2) = $\sum^{N}{}_{I=1} n_i p_i$ = 0,3*2+0,2*2+0.2*2+0,15*3 +0,1*4+0.05*4 = 2,45

Среднее количество информации на один символ первичного алфавита равно

$I^A = - \sum^{N}{}_{I=1} P_i \log P_i$ = 2,39 бит.

Теперь по известной нам формуле найдем избыточность кода Шеннона–Фано.

Q(Шеннона-Фано, А, 2) = К(Шеннона-Фано, А, 2) / I^A – 1 = = 2,45/2,39 – 1 ~ 0,025.

То есть избыточность кода Шеннона-Фано для нашего игрушечного алфавита составляет всего около 2,5 %.

Для русского алфавита этот избыточность кодирования кодом Шеннона-Фано составила бы примерно 0,0147.