

Лекция 12. Алгоритмы.

Часть 1. Алгоритмы теории информации.

Код Хаффмана.

На прошлой лекции был рассмотрен алгоритм Шеннона-Фано, обеспечивающий достаточно экономное кодирование. Еще более экономное (и даже максимально экономное) кодирование удастся осуществить при использовании алгоритма Хаффмана. Построение этого кода опирается на простое преобразование того алфавита, на котором записываются передаваемые по линии связи сообщения, называемое *сжатием* алфавита. Пусть мы имеем алфавит A , содержащий буквы a_1, a_2, \dots, a_n , вероятности появления которых в сообщении соответственно равны p_1, p_2, \dots, p_n , при этом мы считаем буквы расположенными в порядке убывания их вероятностей (или частот).

Условимся теперь не различать между собой две наименее вероятные буквы нашего алфавита, т. е. будем считать, что a_{n-1} и a_n — это одна и та же буква b нового алфавита A_1 , содержащего, очевидно, буквы a_1, a_2, \dots, a_{n-2} и b (т.е. a_{n-1} или a_n), вероятности появления которых в сообщении соответственно равны p_1, p_2, \dots, p_{n-2} и $p_{n-1} + p_n$. Алфавит A_1 и называется полученным из алфавита A с помощью сжатия (или однократного сжатия).

Прилагательное «однократное» в скобках в конце последней фразы имеет следующий смысл. Расположим буквы нового алфавита A_1 в порядке убывания их вероятностей и подвергнем сжатию алфавит A_1 ; при этом мы приходим к алфавиту A_2 , про который естественно сказать, что он получается из первоначального алфавита A с помощью двукратного сжатия (а из алфавита A_1 — с помощью однократного сжатия). Ясно, что новый алфавит A_2 будет содержать уже всего $n-2$ буквы. Продолжая эту процедуру, мы будем приходить ко все более коротким алфавитам; после $n-2$ -кратного сжатия мы приходим к алфавиту A_{n-2} , содержащему уже всего две буквы.

Вот, например, как преобразуется с помощью последовательных сжатий алфавит, содержащий 6 букв, вероятности которых равны 0,4, 0,2, 0,2, 0,1, 0,05 и 0,05.

№ буквы	Вероятности					
	исходный алфавит А	сжатые алфавиты				
		А ₁	А ₂	А ₃	А ₄	
1	0,4	0,4	0,4	0,4	→0,6 0,4	
2	0,2	0,2	0,2	→0,4		
3	0,2	0,2	0,2	0,2		
4	0,1	0,1	→0,2	→0,2		
5	0,05	→0,1	→0,2	→0,2		
6	0,05					

Условимся теперь приписывать двум буквам последнего алфавита A_{n-2} кодовые обозначения 1 и 0. Далее, если кодовые обозначения уже приписаны всем буквам алфавита A_j , то буквам «предыдущего» алфавита A_{j-1} (где, разумеется, $A_{1-1} = A_0$ — это исходный алфавит A , сохранившимся и в алфавите A_j , мы припишем те же кодовые обозначения, которые они имели в алфавите A_{j-1} ; двум же буквам a' и a'' алфавита A_j , «слившимся» в одну букву b алфавита A_{j-1} , мы припишем обозначения, получающиеся из кодового обозначения буквы b добавлением цифр 1 и 0 в конце:

№ буквы	вероятности и кодовые обозначения					
	исходный алфавит A	сжатые алфавиты				
		A ₁	A ₂	A ₃	A ₄	
1	0,4 0	0,4 0	0,4 0	0,4 0	→ 0,6 1 0,4 0	
2	0,2 10	0,2 10	0,2 10	→ 0,4 11		
3	0,2 111	0,2 111	0,2 111	0,2 10		
4	0,1 1101	0,1 1101	→ 0,2 110	→ 0,2 10		
5	0,05 11001	→ 0,1 1100	→ 0,2 110	→ 0,2 10		
6	0,05 11000					

Легко видеть, что из самого построения получаемого таким образом кода Хаффмана вытекает, что он удовлетворяет общему условию: **никакое кодовое обозначение не является здесь началом другого, более длинного кодового обозначения** и поэтому, как и код Шеннона-Фано, всегда декодируется однозначно.

Избыточность.

В естественных языках надежность обеспечивается тем, что существует некоторый компромисс между экономностью и надежностью кода (как записанного буквами, так и переданного фонетически — звуками). Можно попытаться измерить, какой процент сообщений в языке передается для обеспечения надежности передачи (возможности однозначного декодирования любого сообщения) даже при наличии сбоях — утрате букв, слов или даже целых предложений и кусков текста.

Первый текст текст: утрачено 10%

МНОГОМЕТРОВАЯ ВО НА НАКРЫВАЕТ ВОЕЙ ХОЛОДНОЙ ТОЛЩЕЙ ОДКУ И Л ДЕЙ. РУКИ
 ЕРТВОЙ ХВАТКОЙ ДЕРЖАТ К ПРОНОВЫЙ ТРОС

Второй текст: 20%

О СТ НИЕ ОБС АН ВКИ В ЛА СЕ ВЫЗВАЛО ПОДОЗРИТЕ НУЮ ШУ ИХ
 А ИОТАЖ СО ЕДНИХ СТРАН

Третий текст: 30%

ОС ЕРЕ ОЗКИ ИВО ТИ МЯ А И ЫР ВО НОГ
 РО ОХ ЕНИЯ ЖЕЛ ЗНО ОРО НЫЕ АГОНЫ

Ясно, что такой способ измерения неточен. Ведь в некоторых текстах иногда можно по одному слову догадаться, о чем идет речь, тогда как в незнакомой области уже утеря нескольких слов или букв создает непреодолимую преграду.

Чтобы сделать постановку задачи более формальной, рассмотрим некоторый алфавит a_1, a_2, \dots, a_n . Первоначально каждой букве присвоим равную частоту $1/n$, что соответствует полному незнанию особенностей языка (обычно алфавит изучаемого языка напечатан на первой странице учебника или самоучителя). Энтропия на один символ здесь составляет $H_0 = \log_2 n$ Ю а значит для передачи каждой буквы требуется в среднем $H_0 = \log_2 n$ двоичных символов. Найдя реальные частоты букв p_1, p_2, \dots, p_n , мы несколько приблизимся к пониманию особенностей

языка. Текст, учитывающий эти частоты, имеет энтропию $H_1 = -\sum_{i=1}^n p_i \log_2 p_i$, и требует,

соответственно H_1 символов на одно сообщение.

Абсолютное уменьшение составляет $H_0 - H_1$ двоичных символов. Эта мера абсолютной избыточности русского текста в двоичном коде (точнее первое приближение к ее оценке). Она показывает, сколько символов являются «лишними» в среднем на одну букву, и могут быть в принципе восстановлены при утрате, исходя только из частот букв. Правда, такая оценка неудобна, ведь обычно в языке используется не двоичный код, а специфический алфавит. В то же время не хотелось бы привязываться к конкретному алфавиту и иметь возможность сравнивать разные языки, например русский и английский, английский и немецкий и т.д. Поэтому разумно перейти к безразмерным величинам в процентах.

Относительное уменьшение составляет $\frac{H_0 - H_1}{H_0} \cdot 100$ процентов, что можно считать мерой

избыточности любого кода (поскольку эта величина является относительной, то уже не зависит от конкретной реализации алфавита и оказывается свойственной именно самому языку).

Более точную оценку избыточности можно получить, составив таблицу двухбуквенных сочетаний $(a_1 a_1), (a_1 a_2), \dots, (a_1 a_n), (a_2 a_2), \dots, (a_n a_n)$ с соответствующими частотами

$p_{11}, p_{12}, \dots, p_{1n}, p_{22}, \dots, p_{nn}$. Рассчитав энтропию $H_2 = -\sum_{i=1}^n \sum_{j=1}^n p_{ij} \log_2 p_{ij}$, можно далее получить

уточненную оценку избыточности $\frac{H_0 - H_2}{H_0} \cdot 100\%$. Можно доказать, что при повторении этого

процесса, и рассмотрении более длинных комбинаций букв (трехбуквенных, четырехбуквенных сочетаний и т.д.) энтропия будет уменьшаться. Содержательно это достаточно очевидно, поскольку рассматривая более длинные комбинации, мы по существу используем информацию о

1. Построение кодов Хэмминга (описание алгоритма кодирования). Разобьем отрезок натуральных чисел $(1, 2, \dots, l)$ на k последовательностей следующим образом: пусть V — произвольное натуральное число $1 \leq V \leq l$ и (V_k, \dots, V_1) — его двоичная запись.

Последовательность 1, 3, 5, 7, 9, ... содержит все числа V с $V_1 = 1$

Последовательность 2, 3, 6, 7, 10, ... содержит все числа V с $V_2 = 1$.

Последовательность 4, 5, 6, 7, 12, ... содержит все числа V с $V_3 = 1$.

.....

Последовательность $2^{k-1}, 2^{k-1} + 1, \dots$ содержит все числа V с $V_k = 1$.

Первыми членами этих последовательностей являются числа $1 = 2^0, 2 = 2^1, \dots, 2^{k-1}$, т. е. степени двойки, причем $2^{k-1} \leq l$, а $2^k \geq l + 1$.

Члены β_i набора $(\beta_1, \beta_2, \dots, \beta_l)$, у которых индекс i принадлежит множеству $(1, 2, \dots, 2^{k-1})$, называются **контрольными членами**, остальные — **информационными**. Легко видеть, что контрольных членов будет k , а информационных $l - k = m$.

Сформулируем теперь правило построения набора $(\beta_1, \beta_2, \dots, \beta_l)$ по набору $(\alpha_1, \alpha_2, \dots, \alpha_m)$. Сначала определяются информационные члены

$$\beta_3 = \alpha_1$$

$$\beta_5 = \alpha_2$$

$$\beta_6 = \alpha_3$$

.....
Таким образом, набор из информационных членов, расположенных в естественном порядке, совпадает с набором $(\alpha_1, \alpha_2, \dots, \alpha_m)$. Далее определяются контрольные члены

$$\beta_1 = \beta_3 + \beta_5 + \beta_7 + \dots \pmod{2}$$

$$\beta_2 = \beta_3 + \beta_6 + \beta_7 + \dots \pmod{2}$$

$$\beta_4 = \beta_5 + \beta_6 + \beta_7 + \dots \pmod{2}$$

.....
Здесь суммирование ведется по последовательностям, построенным выше. В этих формулах правые части, очевидно, состоят из информационных членов, которые нами уже определены.

II. Обнаружение ошибки в кодах Хэмминга. Пусть при передаче кода $(\beta_1, \beta_2, \dots, \beta_l)$ произошла ошибка в S -м члене. Тогда на выходе канала было принято слово $(\beta'_1, \beta'_2, \dots, \beta'_l)$, где

$$\beta'_1, \beta'_2, \dots, \beta'_l = \beta_1, \dots, \beta_S, \dots, \beta_l$$

Пусть $S = S_k \dots S_1$ — запись числа S в двоичном счислении. Покажем, как можно по коду $(\beta'_1, \beta'_2, \dots, \beta'_l)$ найти число S . Рассмотрим число $S' = S'_k \dots S'_1$, где:

$$S'_1 = \beta'_1 + \beta'_3 + \beta'_5 + \beta'_7 + \dots \quad (1\text{-я последовательность})$$

$$S'_2 = \beta'_2 + \beta'_3 + \beta'_6 + \beta'_7 + \dots \quad (2\text{-я последовательность})$$

$$S'_3 = \beta'_4 + \beta'_5 + \beta'_6 + \beta'_7 + \dots \quad (3\text{-я последовательность})$$

.....

Утверждается, что $S = S'$. В самом деле, если $S_1 = 0$, то S не принадлежит 1-й последовательности и тогда

$$\beta'_1 + \beta'_3 + \beta'_5 + \beta'_7 + \dots = \beta_1 + \beta_3 + \beta_5 + \beta_7 + \dots = 0$$

поэтому $S'_1 = 0$; если $S_1 = 0$, то S принадлежит 1-й последовательности и тогда

$$\beta'_1 + \beta'_3 + \beta'_5 + \beta'_7 + \dots = 1 + \beta_1 + \beta_3 + \beta_5 + \beta_7 + \dots = 1$$

поэтому $S'_1 = 1$. Таким образом, $S_1 = S'_1$.

Аналогично доказывается, что $S_2 = S'_2; \dots; S_k = S'_k$. Отсюда следует, что $S = S'$.

Если при передаче ошибки не произошло, то, очевидно, $S' = 0$. Значит число S' позволяет узнать, произошла ли ошибка при передаче и, если произошла, то найти номер члена S , который искажился помехой. В последнем случае производим коррекцию ошибки: член β'_S заменяем на $\bar{\beta}'_S$.

III. Декодирование. Этот шаг состоит в построении исходного сообщения $(\alpha_1, \alpha_2, \dots, \alpha_m)$ по коду $(\beta_1, \beta_2, \dots, \beta_l)$. Для этого, очевидно, достаточно взять информационные члены в $(\beta_1, \beta_2, \dots, \beta_l)$.

Пример. Построить самокорректирующийся код для $m=4$. Наименьшее число l , удовлетворяющее неравенству

$$2^4 \leq \frac{2^l}{l+1}$$

будет $l=7$, и тогда $k=3$. В соответствии с этапом I получаем самокорректирующийся код. Результат этого построения представлен в таблице, в которой контрольные члены помечены звездочкой.

1*	2*	3	4*	5	6	7
0	0	0	0	0	0	0
1	1	0	1	0	0	1
0	1	0	1	0	1	0
1	0	0	0	0	1	1
1	0	0	1	1	0	0
0	1	0	0	1	0	1
1	1	0	0	1	1	0
0	0	0	1	1	1	1
1	1	1	0	0	0	0
0	0	1	1	0	0	1
1	0	1	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	1	0	1	0	1
0	0	1	0	1	1	0
1	1	1	1	1	1	1

В этой таблице сначала в столбцы с номерами 3, 5, 6 и 7 (информационные члены) вписываются сверху вниз наборы 0000, ..., 1111. Затем по формулам

$$\beta_1 = \beta_3 + \beta_5 + \beta_7 \pmod{2}$$

$$\beta_2 = \beta_3 + \beta_6 + \beta_7 \pmod{2}$$

$$\beta_4 = \beta_5 + \beta_6 + \beta_7 \pmod{2}$$

заполняются столбцы с номерами 1, 2 и 4.

Пусть на вход канала поступил код 0110011, и в нем источник помех искажил 5-й член ($S=5$). Тогда на выходе мы получим 0110111. Вычислим номер члена, в котором произошла ошибка. Мы имеем

$$S'_1 = \beta'_1 + \beta'_3 + \beta'_5 + \beta'_7 = 0 + 1 + 1 + 1 = 1$$

$$S'_2 = \beta'_2 + \beta'_3 + \beta'_6 + \beta'_7 = 1 + 1 + 1 + 1 = 0$$

$$S'_3 = \beta'_4 + \beta'_5 + \beta'_6 + \beta'_7 = 0 + 1 + 1 + 1 = 1$$

Следовательно, $S' = 101$, т. е. $S' = 5$. Мы обнаружили член, в котором произошла ошибка и $S' = S$. Внеся корректировку в 5-й член, получим правильный исходный код 0110011.

ЗАДАЧКА НА ДОМ

Вести из роддома. Жена нового русского в связи со сложным случаем (многоплодная беременность) отправилась рожать в Калифорнию. Сообщение о количестве родившихся наследников должно прийти по спутниковому каналу связи, причем новый русский для надежности заказал передачу с исправлением одиночных ошибок по методу Хэмминга. Двоичный канал передает числа от 0 до 15 в обычной двоичной кодировке (0 – 0000, 1 – 0001, 2 – 0010, ..., 15 – 1111) с добавлением необходимого количества контрольных символов. Однако система автоматического исправления ошибок отказала, поэтому придется исправлять ошибку вручную (если она есть). На выход канала поступило сообщение 1100011. Помогите конкретно человеку узнать, сколько наследников ему родила жена; заодно определите, стоило ли переплачивать за повышенную надежность кода.