Теория информации

Лекция 5

Символьные коды

В данной лекции мы рассмотрим символьные коды переменной длины. Эти коды кодируют за раз один символ, в отличие от блочных кодов рассмотренных до этого, которые кодировали за раз последовательность из N символов. Символьные коды призваны осуществлять компрессию и декомпрессию (или сжатие и распаковку) вообще *без потерь*, то есть гарантировать восстановление точное для всех без исключения символов.

Идея символьных кодов сжатия в том, что можно достичь сжатия в среднем, присваивая короткие коды более вероятным символам и длинные коды менее вероятным. Ключевыми вопросами здесь являются следующие:

- Каковы следствия того, что коды без потерь и если некоторые кодовые слова короткие, то насколько удлинятся остальные?
- Как практически сделать подобные коды пригодными для декодирования?
- Как следует определять длину кодовых слов, чтобы достичь максимального сжатия?

Необходимо снова проверить фундаментальное значение Шенноновского количества информации доказав *теорему кодирования для символьных кодов*.

Теорема. Существует способ кодирования с кодами переменной длины C ансамбля X, такой что средняя длина закодированного символа (или *стоимость кодирования*) $L(C,X) \in \left[H(X),\, H(X)+1\right)$ (или если записать в виде неравенства $H(X) \le L(C,X) < H(X)+1$).

Введем некоторые обозначения. Обозначим A^N множество *N*-кортежей или упорядоченных элементов элементов из алфавита A длиной N. Проще говоря, все строки из символов алфавита A длиной N. Например, если $A=\{0,1\}$, то $A^3=\{0,1\}^3=\{000,001,010,011,100,101,111\}$. Еще обозначим A^+ множество *всех* строк конечной длины, составленных из алфавита A. Например, $\{0,1\}^+=\{0,1,00,01,10,11,000,001,...\}$.

Рассмотрим теперь собственно символьные коды. *Двоичный символьный код* (или просто символьный код) C для ансамбля X это отображение всего диапазона x или $A_x=\{a_1,a_2,...,a_I\}$ на множество $\{0,1\}^+$. Обозначим $c(x)$ кодовое слова соответствующее x, $l(x)$ - длина этого слова, $l_i=l(a_i)$.

Расширенный код C^+ это отображение $A_x^+ \to \{0,1\}^+$, полученное

объединением (конкатенацией) соответствующих кодовых слов, т.е. $c^+(x_1x_2...x_N) = c(x_1)c(x_2)...c(x_N)$ (справа не умножение а просто конкатенация).

Пример:

$$A_x = \{a, b, c, d\} \quad , \tag{5.1}$$
$$P_x = \left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right\} \quad ,$$

символьный код для этого ансамбля C_0 представлен в таблице:

a_i	a	b	c	d
$c(a_i)$	1000	0100	0010	0001
l_i	4	4	4	4

Используя расширенный код кода C_0 строку $acdbac$ можно закодировать как

$$c+(acdbac) = 100000100001010010000010 \quad .$$

Основные требования к символьному коду можно сформулировать следующим образом:

1) каждая закодированная строка должна декодироваться единственным образом;
2) процедура декодирования должна быть простой;
3) код должен обеспечивать максимальное сжатие.

Определение: код $C(X)$ является *однозначно декодируемым* (или *разделимым*), если для его расширения $C^+(X)$ никакие две различные строки не имеют одинаковой кодировки, т.е.

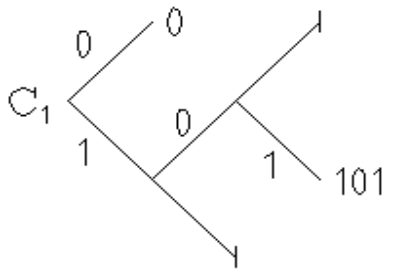
$$\forall \boldsymbol{x}, \boldsymbol{y} \in A_x^+, \ \boldsymbol{x} \neq \boldsymbol{y} \Rightarrow c^+(\boldsymbol{x}) \neq c^+(\boldsymbol{y}) \quad . \tag{5.2}$$

Символьный код легко раскодировать, если, несмотря на то, что кодовые слова могут быть разной длины, конец слова определяется немедленно, как только слово прочитано. Это, в свою очередь, означает, что никакое кодовое слово не может быть *префиксом* другого кодового слова. То есть если в код входит слово. Например код, состоящий из кодовых слов 0, 10, 11 удовлетворяет этому требованию, т.к. сообщение 01001101110 разбивается на слова единственным образом, 0 10 0 11 0 11 10. А вот код, состоящий из кодовых слов 0, 10, 11, 100 уже не удовлетворяет, поскольку сообщение можно трактовать несколькими способами. Так сообщение 01001101110 можно трактовать как последовательность кодовых слов 0 10 0 11 0 11 10 или как

последовательность 0 100 11 0 11 10.

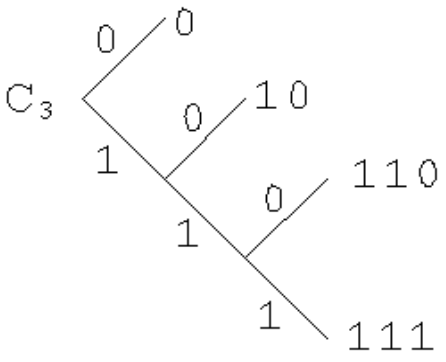
Определение: символьный код называется *префиксным кодом*, если ни одно кодовое слово не является префиксом другого кодового слова. Такой код еще называется *мгновенным* (в англоязычной литературе еще встречаются названия *self-punctuating* или *prefix-free*).

Префиксному коду можно сопоставить дерево. Рассмотрим пару примеров. Коду $C_1=\{0,101\}$ соответствует дерево



.

Например, код $C_2=\{1,101\}$ не префиксный, поскольку 1 это префикс 101. Еще рассмотрим префиксный код $C_3=\{0,10,110,111\}$ и его дерево



.

Из приведенных рисунков видно, что дерево кода C_1 , в отличие от кода C_3 имеет неиспользованные ветви. Префиксный код называется *завершенным префиксным кодом*, когда в соответствующем ему дереве нет неиспользованных ветвей.

Определение. Средняя длина $L(C,X)$ символьного кода C для ансамбля X определяется как

$$L(C,X) = \sum_{x\in A_x} p(x)l(x) = \sum_{i=1}^{I} p_i l_i, \quad I=|A_x| \quad . \tag{5.3}$$

Рассмотрим для примера тот же ансамбль (5.1), воспользуемся для него кодом $C_3=\{0,10,110,111\}$ и сведем все величины в таблицу:

a_i	$c(a_i)$	p_i	$h(p_i)$	l_i
a	0	1/2	1.0	1
b	10	1/4	2.0	2
c	110	1/8	3.0	3
d	111	1/8	3.0	3

Вычислим для этого примера энтропию и среднюю длину кода:

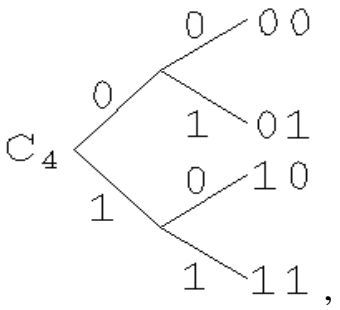
$$H(X) = \sum_i p_i \log_2 \frac{1}{p_i} = 1.75 \ , \tag{5.4}$$

$$L(C_3, X) = \sum_{i=1}^{4} p_i l_i = 1.75 \ . \tag{5.5}$$

Заметим, что в данном случае

$$h(p_i) = \log_2 \frac{1}{p_i} = l_i \quad \text{или} \quad p_i = 2^{-l_i} \ . \tag{5.6}$$

Если бы мы воспользовались для этого ансамбля другим кодом, например $C_4=\{00,01,10,11\}$ с деревом



то средняя длина $L(C_4, X)=2$, что несколько хуже чем для C_3 .

Рассмотрим ограничения накладываемые требованием однозначности декодирования или *разделимости кодирования*. Пусть задан произвольный набор положительных целых чисел $\{l_i\}$. Существует ли разделимый код, для которого этот набор является длинами кодовых слов? Например, мы имеем код $C_4=\{00,01,10,11\}$ и хотим его усовершенствовать заменив 00 на 0, тогда, чтобы сохранить его разделимость или оставить его префиксным, придется удлинить

остальные слова. Таким образом, приходим к идее, что имеется некоторый ограниченный бюджет всевозможных кодовых слов, и укорочение одних слов может привести в целом к увеличению стоимости кодирования $L(C,X)$. Допустим для примера мы используем $l=3$, всего имеется в этом случае $2^3=8$ кодовых слов. Пусть пытаясь усовершенствовать код, мы добавили 0 к набору кодовых слов. Однако из-за этого придется убрать все слова начинающиеся на 0 и 00. Тогда в наборе останется $\{0,100,101,110,111\}$, т.е. вместо 8 слов в наборе останется всего 5. Можно рассуждать и иначе: добавление кодового слова 0 длиной 1 выбросило из набора 4 кодовых слова длиной 3. Таким образом можно сказать, что вес слова длиной 3 в 2^2 раз меньше, чем вес слова длиной 1.

Если задавать полный бюджет кодовых слов равным единице, а каждому кодовому слову приписывать вес 2^{-l} , где l - длина этого слова, то получим некоторую систему оценки кода. Если все слова имеют длину $l_i=3$, то их цена (вес) $2^{-3}=1/8$, а всего их 8 и бюджет $\sum_i 2^{-l}=1$. То же самое имеет место для двух слов длиной 1. Таким образом, приходим к соотношению, называемому неравенством Крафта (или Крафта-Макмилана).

Неравенство Крафта

Теорема. Для любого однозначно декодируемого (разделимого) кода $C(X)$ отображающего произвольный алфавит A_x на двоичный алфавит $\{0,1\}$, длины кодовых слов должны удовлетворять неравенству

$$\sum_{i=1}^{I} 2^{-l_i} \le 1 \ , \qquad (5.7)$$

где $I=|A_x|$.

Если выполняется точное равенство, то речь идет о завершенном коде. В доказательстве сосредоточимся только на префиксном коде, поскольку считается, то он оптимальный.

Доказательство. Определим $S=\sum_{i=1}^{I} 2^{-l_i}$ и рассмотрим величину

$$S^N=\left[\sum_i 2^{-l_i}\right]^N=\sum_{i_1=1}^{I}\sum_{i_2=1}^{I}\dots\sum_{i_N=1}^{I} 2^{-(l_{i_1}+l_{i_2}+\dots+l_{i_N})} \ . \qquad (5.8)$$

Величина $(l_{i_1}+l_{i_2}+\dots+l_{i_N})$ это длина кодирующей последовательности для строки $\boldsymbol{x}=a_{i_1}a_{i_2}\dots a_{i_N}$. Для каждой строки $\boldsymbol{x}$ длиной N имеется только одно слагаемое в сумме (5.8). Если ввести коэффициенты A_l , которые учитывают

сколько раз встречается строка длиной l (это все равно, что привести подобные в (5.8)), то (5.8) можно представить как

$$S^N = \sum_{l=Nl_{min}}^{Nl_{max}} 2^{-l} A_l \ , \qquad (5.9)$$

где $l_{max} = max\ l_i$ и $l_{min} = min\ l_i$. Для строки с длиной кодового слова l может быть всего 2^l кодовых слов, следовательно, 2^l это оценка сверху коэффициента A_l , т.е. $A_l \le 2^l$. Тогда

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

Итак, $S^N \le N l_{max}$. Если бы S была бы >1 , то для достаточно больших N величина S^N обязательно стала бы больше чем Nl_{max} . Однако мы получили, что $S^N \le Nl_{max} \ \forall \ N$, следовательно $S \le 1$. #

Рассмотрим теперь на какое сжатие можно рассчитывать используя префиксные (и не префиксные тоже) коды. Что надо сделать, чтобы минимизировать стоимость кодирования $L(C,X)=\sum_i p_i l_i$? Как мы уже отмечали, энтропия ансамбля $H(X)$ это нижняя граница средней длины кодового слова $L(C,X)$. Докажем это простое утверждение.

Теорема. Стоимость кодирования $L(C,X)$ ограничена снизу энтропией $H(X)$.

Доказательство. Определим следующие вероятности: $q_i \equiv \frac{2^{-l_i}}{z}$, где $z=\sum_{i'} 2^{-l_{i'}}$. Тогда $l_i=\log_2\frac{1}{q_i}-\log_2 z$. Воспользуемся неравенством Гиббса (2.17), $D_{KL}(P||Q)=\sum_i p_i \log_2 \frac{p_i}{q_i} \ge 0 \ \Rightarrow \ \sum_i p_i \log_2 \frac{1}{q_i} \ge \sum_i p_i \log_2 \frac{1}{p_i}$ и неравенством Крафта $z \le 1$, что позволит записать

$$L(C,X)=\sum_i p_i l_i=\sum_i \left(p_i \log_2 \frac{1}{q_i} - p_i \log_2 z \right)=\sum_i p_i \log_2 \frac{1}{q_i} - \log_2 z \ge \sum_i p_i \log_2 \frac{1}{p_i} - \log_2 z \ge H(X) \ .$$

Равенство выполняется тогда, когда выполняется равенство в неравенстве Крафта, т.е. $z=1$ и длины кодовых слов таковы, что $l_i=\log_2\frac{1}{p_i}$. #

Итак, стоимость кодирования $L(C,X)$ минимальна и равна $H(X)$ если длины кодовых слов равны Шенноновскому количеству информации, т.е. $l_i=\log_2\frac{1}{p_i}$. С другой стороны, мы задали распределение связанное с длинами кодовых слов $q_i=\frac{2^{-l_i}}{z}$. В оптимальном завершенном префиксном коде $z=1$, поэтому в этом случае $q_i=2^{-l_i}$. Ясно, что осуществить сжатие ниже порога $H(X)$ нельзя. Но насколько можно приблизиться к этому порогу? Докажем теорему кодирования для символьных кодов, которая приведена в начале лекции. Собственно формулировка теоремы это неравенство

$$H(X)\le L(C,X)<H(X)+1 \quad .$$

Доказательство. Поскольку левую часть неравенства мы уже доказали, осталось доказать правое неравенство. Установим длину кодового слова немного больше оптимальной, а именно зададим

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

где скобки $\lceil\rceil$ означают ближайшее целое число сверху, поскольку числа l_i все равно должны быть целыми. Проверим выполнение неравенства Крафта для такого кода:

$$\sum_i 2^{-l_i}=\sum_i 2^{-\lceil\log_2\frac{1}{p_i}\rceil}\le\sum_i 2^{-\log_2\frac{1}{p_i}}=\sum_i p_i=1 \quad .$$

Неравенство выполняется, значит существует префиксный код с заданными l_i . Теперь оценим сверху стоимость кодирования:

$$L(C,X)=\sum_i p_i\lceil\log_2\frac{1}{p_i}\rceil<\sum_i p_i\left(\log_2\frac{1}{p_i}+1\right)=H(X)+1 \quad .\#$$