Код Гилберта-Мура

5 января 2023 г. 1:22

Проблема кода Шеннона: если символов очень много (близко к ∞), то сортировка вероятностей усложняет всю процедуру
Мы имеем побуквенный код и стремимся расширить алфавит для улучшения сжатия, т.е. пытаемся приблизиться к наименьшей возможной избыточности. Код Шеннона такую возможность не дает, т.к. требует сортировки
Если не сортировать, получится шляпа:

x	$p(x)$	$q(x)$	$l(x)$	$c(x)$
a	0.1	0	4	0000...
b	0.3	0.1	2	00...
c	0.6	0.4	1	0...

не декодируется однозначно

Используя код Гилберта-Мура, можно избежать необходимость сортировки вероятностей за счет введения дополнительной избыточности
Результат для кода Гилберта-Мура:

x	$p(x)$	$q(x)$	$\sigma(x)$	$l(x)$	$c(x)$
0	0.1	0.0	0.05	5	00001
1	0.6	0.1	0.4	2	01
2	0.3	0.7	0.85	3	110

Код Гилберта-Мура сжимает еще хуже, чем код Шеннона, зато не требует сортировки

Итак, в коде Гилберта-Мура не происходит сортировки вероятностей и вводится модифицированная кумулятивная вероятность.
Т.е. имеем ансамбль $X = \{1, 2, \dots, M\}$, $\{p_1, p_2, \dots, p_M\}$
Для каждого $x \in X$ вычислим модифицированную кумулятивную вероятность (сигма):
$\sigma_i = q_i + \frac{p_i}{2}$, $i = 1, 2, \dots, M$
Здесь $q_1 = 0$, $q_i = \sum_{j=1}^{i-1} p_j$. Кодовое слово x_i - это двоичная последовательность, представляющая собой первые $l_i = \left\lceil -log\left(\frac{p_i}{2}\right) \right\rceil$ бит после запятой в двоичном представлении σ_i
Т.е. длина кодового слова будет больше на 1, чем в коде Шеннона (средняя длина кодового слова < H+2 для кода Гилберта-Мура)

Пример:

x	$p(x)$	$q(x)$	$\sigma(x)$	$l(x)$	$c(x)$
a	0.35	0	0.175	3	001...
b	0.20	0.35	0.450	4	0111...
c	0.15	0.55	0.625	4	1010...
d	0.1	0.70	0.750	5	11000...
e	0.1	0.80	0.850	5	11011...
f	0.1	0.90	0.950	5	11110...

т.е. этот код не следует использовать для побуквенного кодирования

$$\bar{l} = \sum_x p(x)l(x) = 3.95 > H = 2.4016$$

Код Гилберта-Мура является префиксным
Код является префиксным: для $i < j, \sigma_j > \sigma_i$

$$\begin{aligned}\sigma_j - \sigma_i &= \sum_{h=1}^{j-1} p_h + \frac{p_j}{2} - \sum_{h=1}^{i-1} p_h - \frac{p_i}{2} = \\ &= \sum_{h=i}^{j-1} p_h + \frac{p_j - p_i}{2} \geq \\ &\geq p_i + \frac{p_j - p_i}{2} = \\ &= \frac{p_j + p_i}{2} \geq \frac{\max\{p_i, p_j\}}{2} \geq 2^{-\min\{l_i, l_j\}},\end{aligned}$$

где

$$l_m = \left\lceil -\log \frac{p_m}{2} \right\rceil \geq -\log \frac{p_m}{2}.$$

Это означает, что слова $\boldsymbol{c}_i$ и $\boldsymbol{c}_j$ различаются в одном из первых $\min\{l_i, l_j\}$ двоичных символов, т.е., ни одно из двух слов не может быть началом другого.

Свойства префиксного кода:

1. Любое кодовое слово не должно быть началом другого
2. Декодирование однозначно (причем однозначно декодируемый код не обязательно префиксный)
3. Кодовым словам соответствуют только листья двоичного кодового дерева
4. Древовидный код является префиксным

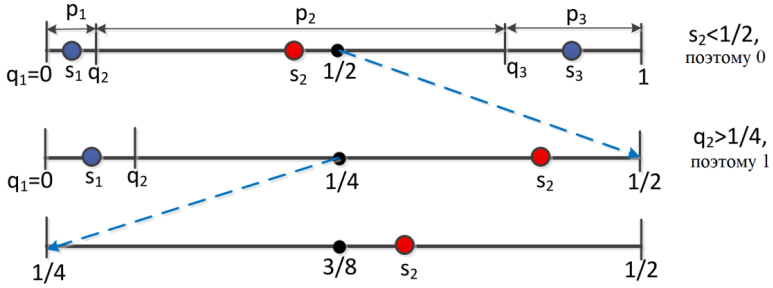
Графическая интерпретация кода Гилберта-Мура:

1. Отмечаем на отрезке от 0 до 1 значения q и σ (на рисунке - q и s)
2. Как и в коде Шеннона, делим отрезок пополам и смотрим, с какой стороны нужная модифицированная кумулятивная вероятность, формируем кодовое слово (справа от середины - 1, иначе 0)
3. Выполняем, пока на отрезке не останется только одна σ - рассматриваемая нами

- $X = \{x_1, x_2, x_3\}$, $p(x_1) = 0.1, p(x_2) = 0.6, p(x_3) = 0.3$
- $q(x_1) = 0, q(x_2) = 0.1, q(x_3) = 0.7$
- $s(x_1) = 0.05, s(x_2) = 0.4, s(x_3) = 0.85$

Рассмотрим кодирование x_2

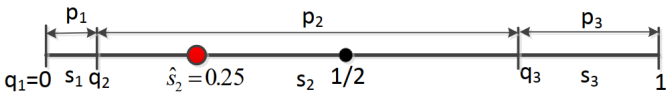
Рассмотрим кодирование x_2:



Декодер:
Декодер получает не сами "сигмы", а их аппроксимацию, потому что вместо 0.4 мы передали 2 бита: 01.
Назовем их округленными "сигмами"
(Если взять 0.01 в бинарном представлении, это 0.25 в десятичной СС)

1. На отрезке от 0 до 1 отмечаем q, "сигмы" и округленные "сигмы"
2. Смотрим, между какими q попала искомая округленная "сигма"
3. Выдаем левую q. Точнее, x, ей соответствующий

- $p(x_1) = 0.1, p(x_2) = 0.6, p(x_3) = 0.3$
- $q(x_1) = 0, q(x_2) = 0.1, q(x_3) = 0.7$
- $s(x_1) = 0.05, s(x_2) = 0.4, s(x_3) = 0.85$
- Декодируем кодовое слово 01 или $\hat{s} = 0.25$.



- После округления до $l(x_i)$ разрядов, число $s(x_i) = q(x_i) + p(x_i)/2$ уменьшается не более чем на $p(x_i)/2$, поскольку ошибка округления не больше, чем $2^{-l(x_i)} \leq p(x_i)/2$.
- Декодирование: найти x_i, такое что $q(x_i) \leq \hat{s} < q(x_{i+1})$.

Итак, код Гилберта-Мура лучше кода Шеннона отсутствием необходимости сортировать вероятности, но хуже с точки зрения избыточности. Чтобы уменьшить кодовую избыточность, можно использовать блоковое кодирование