Арифметическое кодирование с адаптивной оценкой вероятностей алгоритмом A

5 января 2023 г. 1:25

Общая идея адаптивного кодирования

- Кодеру не доступны сообщения, которые появятся в будущем, т.е., при кодировании x_i, сообщения $x_{i+1}, x_{i+2}, ...$ считаются неизвестными.
- По последовательности уже закодированных сообщений $x_0, x_1, ..., x_{i-1}$ кодер оценивает вероятность для символа x_i и строит для него код в соответствии с этой оценкой.
- После декодирования сообщений $x_0, x_1, ..., x_{i-1}$ декодер оценивает вероятность для символа x_i так же как и кодер, после чего декодирует x_i.

Пусть необходимо передать $\boldsymbol{x} = (x_1, ..., x_n)$ арифметическим кодером. Для этого каждому символу x_t необходимо сопоставить $\hat{p}_t(a)$ – оценку вероятности того, что $x_t = a, a = 1, ..., M$. Предположим, что $x_1, ..., x_{t-1}$ уже переданы и известны декодеру. Тогда

$$\hat{p}_t(a) = \frac{\tau_t(a)}{t}$$, где $\tau_t(a)$ число сивмолов a в $x_1, ..., x_{t-1}$. t - сколько всего символов уже было закодировано

$$\hat{p}_t(a) = \frac{\tau_t(a)+1}{t+M}$$, поправка, чтобы избежать нулевых вероятностей.

$$\hat{p}_t(a) = \frac{\tau_t(a)+1/2}{t+M/2}$$. – второй вариант поправки

закодированные ранее (до a) символы
когда на старте все счетчики = 0, p = 0, поэтому нужна поправка
Т.к. из-за нее все p будут смещены, мы начнем проигрывать в сжатии
Со вторым вариантом поправки этот проигрыш меньше

Алгоритмы A и D - в них не надо использовать смещение вероятностей, описанное выше

- Можно использовать подход основанный на так называемом esc-символе.
- В этом случае, мы добавляем дополнительный символ в алфавит. Этот символ передаётся, если на вход приходит символ, который ранее не появлялся.

Общая идея:

- Используется оценка $p_t(a) = \frac{\tau_t(a)}{t+1}$, если $\tau_t(a) > 0$
- Передаётся “esc”, если $\tau_t(a) = 0$, $p_t(esc) = \frac{1}{t+1}$

t+1, т.к. добавился esc, мы считаем, что он всегда встречался в строке 1 раз. Т.е. t символов алфавита и +1 esc-символ было закодировано

Алгоритм A:

закодировали esc

$$\hat{p}_t(a) = \begin{cases} \frac{\tau_t(a)}{t+1}, & \text{если } \tau_t(a) > 0; \\ \frac{1}{t+1}\frac{1}{M-M_t}, & \text{если } \tau_t(a) = 0, \end{cases}$$

закодировали новый символ (появившийся впервые) не $\frac{1}{M}$, т.к. кол-во неизвестных букв постоянно сокращается, выгоднее $\frac{1}{M-M_t}$ – тратится меньше бит

M_t – число различных символов, встретившихся в последовательности длины t.

- В Алгоритме A появление *esc* символа оценивается с меньшей вероятностью, чем это происходит на начальном этапе кодирования. Поэтому, имеет смысл модифицировать оценки вероятностей так, чтобы увеличить вероятность $p_t(esc)$.

Это решает алгоритм D

Когда M_t достигнет M, можно избавиться от esc-символа и кодировать без него

Theorem

При кодировании дискретного постоянного источника с энтропией H, средняя скорость адаптивного арифметического кодирования удовлетворяет неравенству

$$\bar{R} \le H + \frac{M}{2}\frac{\log(n+1)+K}{n},$$

где K не зависит от длины последовательности n.

при $n \to \infty$ оценка вероятностей будет ближе к модели источника, мы приблизимся к энтропии