

Арифметическое кодирование с адаптивной оценкой вероятностей алгоритмом D

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}, \dots считаются неизвестными.
- По последовательности уже закодированных сообщений x_0, x_1, \dots, 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, \dots, x_{i-1} декодер оценивает вероятность для символа x_i так же как и кодер, после чего декодирует x_i .

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

$$\hat{p}_t(a) = \frac{\tau_t(a)}{t}, \text{ где } \tau_t(a) \text{ число символов } a \text{ в } x_1, \dots, x_{t-1}.$$

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

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

закодированные ранее (до a) символы
когда на старте все счетчики = 0, $p=0$,
потому нужна поправка
Т.к. у-да не все p будут смещены,
мы начнем править в счетчик
со вторым вариантом поправки
этот вариант 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 передается, если на вход приходит символ, который ранее не появлялся.

Общая идея:

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

- 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 $p_t(\text{esc})$.

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

→ проблема алгоритма A,
которая решена в D.

Алгоритм D:

$$\hat{p}_t(a) = \begin{cases} \frac{\tau_t(a)-1/2}{t}, & \text{если } \tau_t(a) > 0; \\ \frac{1}{M}, & \text{если } \tau_t(a) = 0, t = 0; \\ \frac{M/2}{2t} \frac{1}{M-M/2}, & \text{если } \tau_t(a) = 0, t > 0. \end{cases}$$

если кодирует
символ, который
уже встречался

кодирует новый символ
→ esc-символ

→ в A тоже $\frac{1}{M}$ (это самое начало,
когда кодируется первый
символ)

Алгоритм D за счет того, что вероятность esc-символа оценивается лучше, чем в алгоритме A, на кодирование esc-символа тратится меньше бит

Theorem

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

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

где K не зависит от длины последовательности n .

