

Алгоритм PPM-A

5 января 2023 г. 1:26

Контекстное адаптивное арифметическое кодирование

Его идея заключается в том, чтобы кодировать каждый символ, учитывая как можно больше предыдущих символов

t - кол-во предыдущих символов, учитываемых алгоритмом (контекст)

- Если распределение вероятностей вида $\hat{p}(x_{t+1}|x_1, \dots, x_t)$ возможно оценить с достаточной точностью, то арифметический кодер может использовать эти вероятности для достижения максимального сжатия, т.е., среднюю длину кодового слова, близкую к $H(X|X^\infty)$.
- Каждое условие соответствует источнику без памяти, к которому может быть применено универсальное кодирование.
- Избыточность адаптивного кодирования приближается к нулю с ростом длины кодируемой последовательности. Поэтому в каждый контекст должно попасть достаточное количество символов, т.е., контекстов не должно быть слишком много.
- Эффективность такого кодирования сильно зависит от метода оценки вероятностей.

индекс
от $t-d+1$
до t

Строка (как бы подстрока входной последовательности) $s = x_{t-d+1}^t$ длины $d \leq D$, предшествующая x_{t+1} , является контекстом для x_{t+1} , если s уже появлялась в x_1^{t-1} .

D - ограничение на память и на вычислительную сложность

Пример определения контекста двух букв (отмеченных красным)

Пример. THE _ CAT _ IN _ THE _ **C** A T E _ THE _ **R** A T

t	буква	контекст
16	A	THE _ C
27	R	THE _

PPM - prediction by partial matching - предсказание по частичному совпадению

Основные этапы кодирования символа x_{t+1} :

- Выполняется поиск контекста $s = x_{t-d+1}^t$ наибольшей длины d , не превышающей D .
 - Для всех возможных значений символа x_{t+1} вычисляются оценки условных вероятностей символа при известном контексте s .
 - Значение символа x_{t+1} кодируется арифметическим кодом в соответствии с вычисленной условной вероятностью.
- D - параметр алгоритма.
 - Вероятность того, что символ $x_{t+1} = a$ после контекста $s = x_{t-d+1}^t$ может быть оценена как

$$\hat{p}_t(a|s) = \frac{\tau_t(s, a)}{\tau_t(s)}.$$

- Если буква для данного контекста не встречалась, т.е. $\tau_t(s, a) = 0$, то арифметический кодер не сможет использовать $\hat{p}_t(a|s) = 0$. Поэтому используется esc-символ.
- Если $\hat{p}_t(a|s) = 0$, то передаётся esc-символ и контекст укорачивается на одну букву.

сколько раз появлялось a
в контексте s

сколько раз этот контекст s
уже появлялся

последовательность данных
была уже до этого встречалась,
тогда она может стать контекстом,
когда $d > 0$

Тот же алгоритм в ином представлении

- Находим наибольшее d такое, что $\hat{p}_t(x_{t-d+1}^t) > 0$, $d \leq D$.
- Выбираем контекст $s \leftarrow x_{t-d+1}^t$ пока эта буква при нашем контексте s не встречалась...
- while** $\hat{p}_t(x_{t+1}|s) = 0$ **do**
- Кодируем esc в соответствии с $\hat{p}_t(\text{esc}|s)$.
- Уменьшаем длину контекста: $d \leftarrow d - 1$, $s \leftarrow x_{t-d+1}^t$.
- end while** контекст найден (хотя бы длиной в 1 символ)
- if** $d > 0$ **then**
- Кодируем x_{t+1} в соответствии с $\hat{p}_t(x_{t+1}|s)$.
- else** контекста не было, но символ появлялся
- if** $\hat{p}_t(x) > 0$ **then**
- Кодируем x_{t+1} в соответствии с $\hat{p}_t(x)$
- else**
- Передаём esc и кодируем x_{t+1} в соответствии с равномерным распределением на не встречавшихся в x_1^t буквах.
- end if**
- end if**

Алгоритм A:

контекст найден

буква a уже встречалась

$$\hat{p}_t(a|\#) = \frac{\tau_t(a)}{t+1}; \quad \hat{p}_t(esc|\#) = \frac{1}{t+1}, \tau_t(a) > 0$$

$$\hat{p}_t(a|s) = \frac{\tau_t(s, a)}{\tau_t(s) + 1}; \quad \hat{p}_t(esc|s) = \frac{1}{\tau_t(s) + 1}, \tau_t(s, a) > 0$$

Принцип "исключений" алгоритма PPM

Предположим, мы нашли контекст s , и оказалось, что для него нужно передавать esc -символ, т.е. ни разу не было буквы a за этим контекстом s .

Передаем esc -символ и сокращаем контекст на 1.

Далее:

После получения esc декодер всё еще не знает x_{t+1} , но он знает, какие символы не могут быть на этой позиции. Это символы, которые следовали за контекстом s , когда он появлялся ранее. Это знание позволяет уточнять вероятность $\hat{p}(x_{t+1}|s') = (s_{d-1}, \dots, s_1)$.

Список исключаемых символов растёт, если esc снова передаётся.

Исключение некоторых символов увеличивает оценку вероятности для оставшихся символов, т.е., увеличивает сжатие.

Пример.

- Рассмотрим контекст $s = "кор"$ за которым следует буква " $т$ ", причем " $корт$ " ранее не встречалось. При этом предположим, что ранее после s встречались " $а$ " и " $с$ ".
- Передаём esc и укорачиваем контекст до $s = "ор"$ *не увеличивает*
- Тогда $\tau'_t(ор) = \tau_t(ор) - \tau_t(ора) - \tau_t(орс)$.