Современные проблемы информатики

Кодирование источника. Оценивание вероятностей

Фионов Андрей Николаевич

СибГУТИ

2020

Оценивание вероятностей через счетчики встречаемости

Каждой букве алфавита $A = \{a_1, \ldots, a_N\}$ сопоставим счетчик встречаемости $C = (c_1, \ldots, c_N)$, $\sum c_i = D$. Тогда

$$\hat{p}(a_i) = c_i / D$$

- начальное значение всех счетчиков – единицы $(D = N)$
- очередной символ кодируем на основании текущих счетчиков
- увеличиваем счетчик закодированного символа

Декодирование: повторяем аналогичные действия

Насколько увеличивать счетчик?

+1 – L-Estimator (Лаплас)

+2 – KT-Estimator (Кричевский–Трофимов) – оптимальный для стационарных эргодических источников

Избыточность KT на символ (источник без памяти)

$$r < \frac{N-1}{2}\frac{\log n}{n} + O(1/n)$$

$$r \to 0 \text{ при } n \to \infty$$

Получаем *универсальное* кодирование (заданная произвольно низкая избыточноть достигается при любом распределении вероятностей символов)

Дважды универсальное кодирование (Б. Я. Рябко)

Кодируем сообщение, используя модели порядка 0, 1, 2 и т.д., выбираем ту модель, которая дает наименьшую длину кодовой последовательности. Сообщаем ее номер декодеру и передаем соответствующий код. Декодер использует для декодирования указанную ему модель.

MDL-принцип (Minimum Description Length): минимизируется *суммарная* длина описания модели и соответствующего ей кода

Прогнозирование временных рядов

Б. Я. Рябко и др.

Прогноз для $t+1$-го символа (значения) = распределение вероятностей, построенное в ходе кодирования предыдущих t символов (значений).

Взвешивание.
Пусть имеются оценки вероятности появления символа u, полученные в моделях разного порядка: $p_0(u)$, $p_1(u)$, $p_2(u)$, ...
Введем веса w_0, w_1, w_2, ..., $\sum w_i = 1$.
Тогда взвешенная оценка

$$P(u) = w_0 p_0(u) + w_1 p_1(u) + w_2 p_2(u) + \cdots$$

Нестационарные источники

Адаптивное кодирование

Скользящее окно: после кодирования очередного символа увеличиваем счетчик его встречаемости и настолько же уменьшаем счетчик встречаемости символа, который выходит за пределы видимости окна (окно сдвигается вправо)

Можно выбрать размер окна так, чтобы сумма счетчиков была равна 2^k. В арифметическом кодере исчезают операции деления.

Проклятие размерности

Пример. Библия на англ. языке: 4,047,392 байта в кодировке ASCII
Для качественного сжатия используем модель порядка 8
Размер модели: $256^8 = 2^{64}$ контекстов по 256 счетчиков

Проблемы:

- бóльшая часть контекстов никогла не встречается
- многие контексты встречаются только несколько раз – оценки вероятностей некачественные
- в частых контекстах встречаются лишь несколько из 256 возможных символов

Контекст _the_hea встречается 644 раза. В нем идут только 6 различных символов: d, l, p, r, t ,v.

Семейство алгоритмов PPM

PPM = Prediction by Partial Matching (Cleary, Witten, Moffat)

Сообщение $x^n = x_1 x_2 \ldots x_n, \quad x_i \in A = \{a_1, a_2, \ldots, a_N\}$

Добавим в алфавит спец. символ $a_{N+1} = \mathrm{esc}$.

Для различных контекстов будем динамически создавать массивы счетчиков вида $C = (c_1, \ldots, c_{N+1})$ (включая счетчик для esc) с *нулевыми* начальными значениями.

Если кодируемый символ сообщения в текущем контексте имеет нулевой счетчик, будем кодировать его на основании счетчиков для контекста меньшей длины, сообщая об этом декодеру путем вставки в кодовую последовательность кода символа esc.

Введем массив счетчиков $U = (1, 1, \ldots, 1, 0)$ для контекста минус первого порядка.

Алгоритм PPM-D

$\mathrm{PPM\_enc}\ (i, \mu)$

(i – номер кодируемого символа в сообщении,

μ – длина контекста)

IF $\mu < 0$ THEN кодируем x_i на основе U, $U[x_i] \leftarrow 0$;

ELSE

Пусть C – массив счетчиков для контекста $x_{i-\mu}, \ldots, x_{i-1}$

IF C не существует THEN

создаем C для этого контекста, $C = (0, 0, \ldots, 0)$;

IF $C[x_i] \neq 0$ THEN кодируем x_i на основе C, $C[x_i] \leftarrow C[x_i] + 2$;

ELSE

IF $C[\mathrm{esc}] \neq 0$ THEN кодируем esc на основе C;

$\mathrm{PPM\_enc}\ (i, \mu - 1)$;

$C[x_i] \leftarrow 1$, $C[\mathrm{esc}] \leftarrow C[\mathrm{esc}] + 1$.

Алгоритм PPM-D

Путь M – максимальная длина контекста

Кодирование сообщения:

1. Передаем длину сообщения n.
2. FOR $i = 1, 2, \ldots, M$ DO $\mathrm{PPM\_enc}\ (i, i-1)$.
3. FOR $i = M+1, \ldots, n$ DO $\mathrm{PPM\_enc}\ (i, M)$.

К О Н Е Ц