Сжатие без учёта контекста. Разделимые и неразделимые коды

Александра Игоревна Кононова

МИЭТ

28 января 2021 г. — актуальную версию можно найти на `https://gitlab.com/illinc/otik`

Энтропийное сжатие

Модель источника X — источник без памяти,
строится по кодируемому сообщению C:

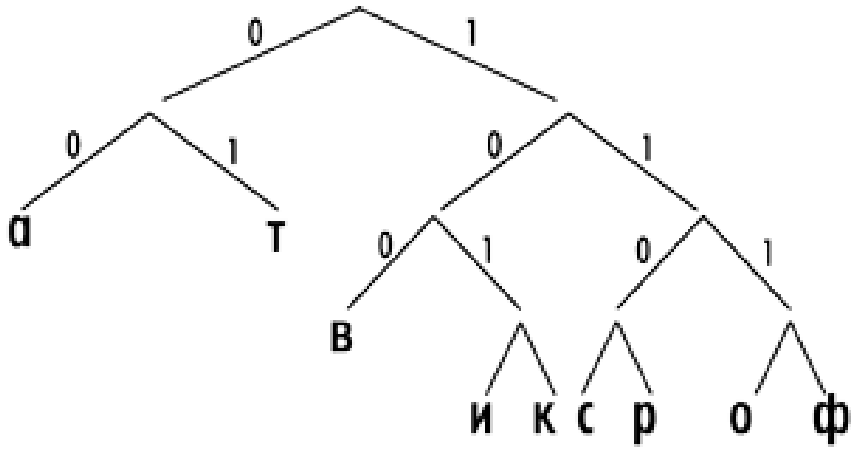
1. кодируемое сообщение — $C \in A_1^+$ (на практике символы первичного алфавита $a \in A_1$ — байты);
2. символы считаются независимыми: $p(a) = const$ (но $p(a_i) \neq p(a_j)$ в общем случае для $a_i, a_j \in A_1$);
3. их вероятности оцениваются по частотам в сообщении C;
4. количество информации $I(a_i)$ (как и суммарное $I(C)$, и среднее источника $I(X)$) оценивается исходя из оценок $p(a_i)$.

Если $\forall a_i, a_j \in A_1$ верно $p(a_i) = p(a_j)$ — модель без памяти X не избыточна, энтропийное сжатие не уменьшит объёма;
если вероятности символов (байтов) не равны друг другу (и $\frac{1}{256}$) — энтропийное сжатие уменьшит объём данных приблизительно до $I(X)$.

Алфавитное префиксное кодирование

1. Каждому символу $a \in A_1$ сопоставляется код $code(a) \in A_2^+$, для двоичного кодирования — $A_2 = \{0, 1\}$ и $code(a)$ — префиксный код из 0 и 1.
2. Длина кода $code(a)$ должна быть как можно ближе к $I(a)$ (для двоичного кодирования — в битах).

Префиксный код = дерево



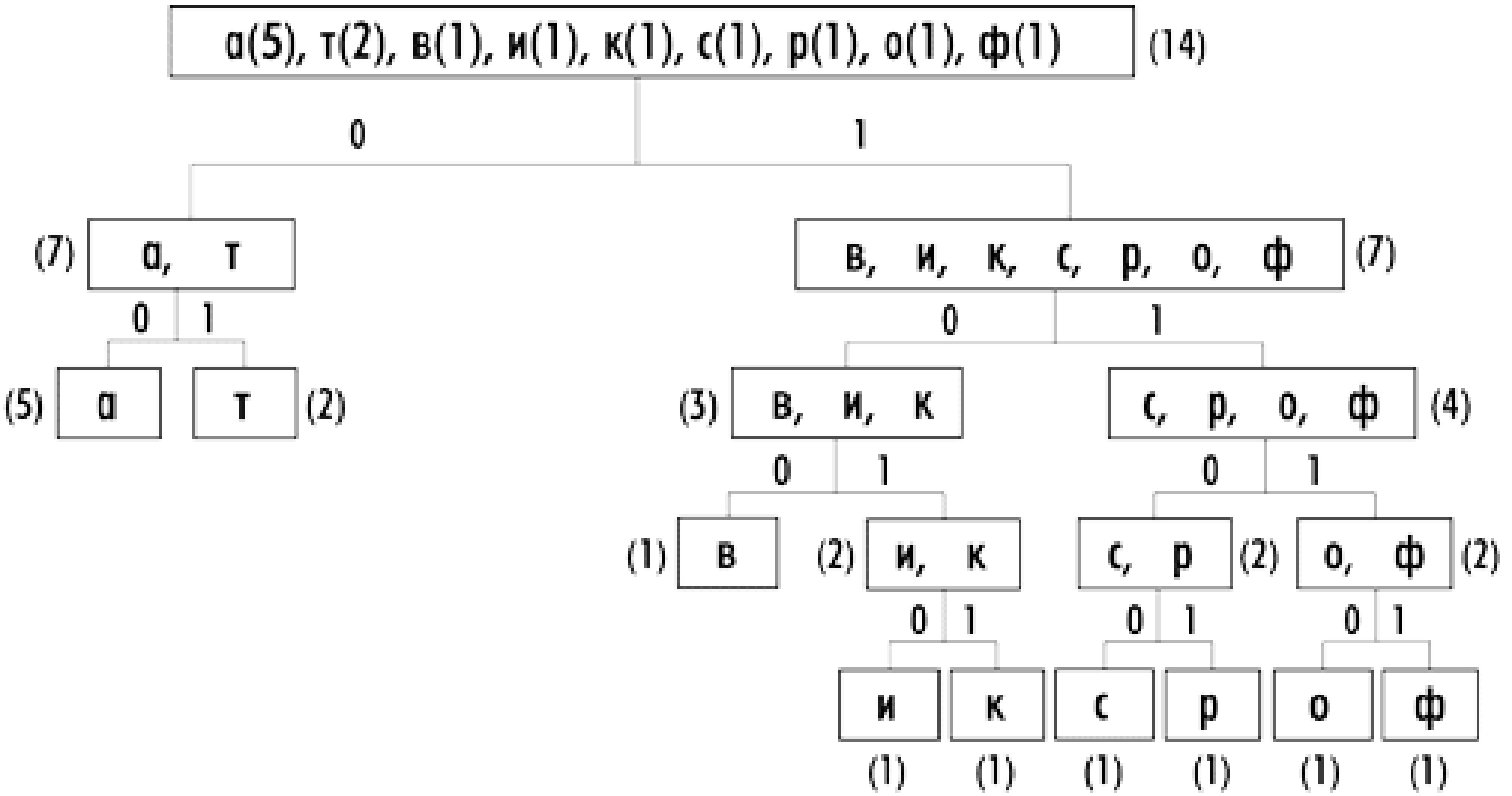
Оптимальный код — сбалансированное с учётом весов дерево.

Код Шеннона–Фано

Дерево Шеннона—Фано строится **сверху вниз**:

1. все символы сортируются по частоте;
2. упорядоченный ряд символов делится на две части так, чтобы в каждой из них сумма частот символов была примерно одинакова;
3. новое деление.

«Авиакатастрофа» — кодирование Шеннона–Фано



а 00
т 01
в 100
и 1010
к 1011
с 1100
р 1101
о 1110
ф 1111

Код Хаффмана

Дерево Хаффмана строится **снизу вверх** (от листовых узлов к корневому узлу):

1. все символы сортируются по частоте;
2. два последних (самых редких) элемента отсортированного списка узлов заменяются на новый элемент с частотой, равной сумме исходных;
3. новая сортировка.

«Авиакатастрофа» — кодирование Хаффмана

{а(5), т(2), в(1), и(1), к(1), с(1), р(1), S1(2)[о(1), ф(1)]}

{а(5), т(2), S1(2), в(1), и(1), S2(2)[с(1), р(1)]}

{а(5), т(2), S1(2), S2(2), в(1), S3(2)[и(1), к(1)]}

{а(5), т(2), S1(2), S2(2), S4(3)[S3(2), в(1)]}

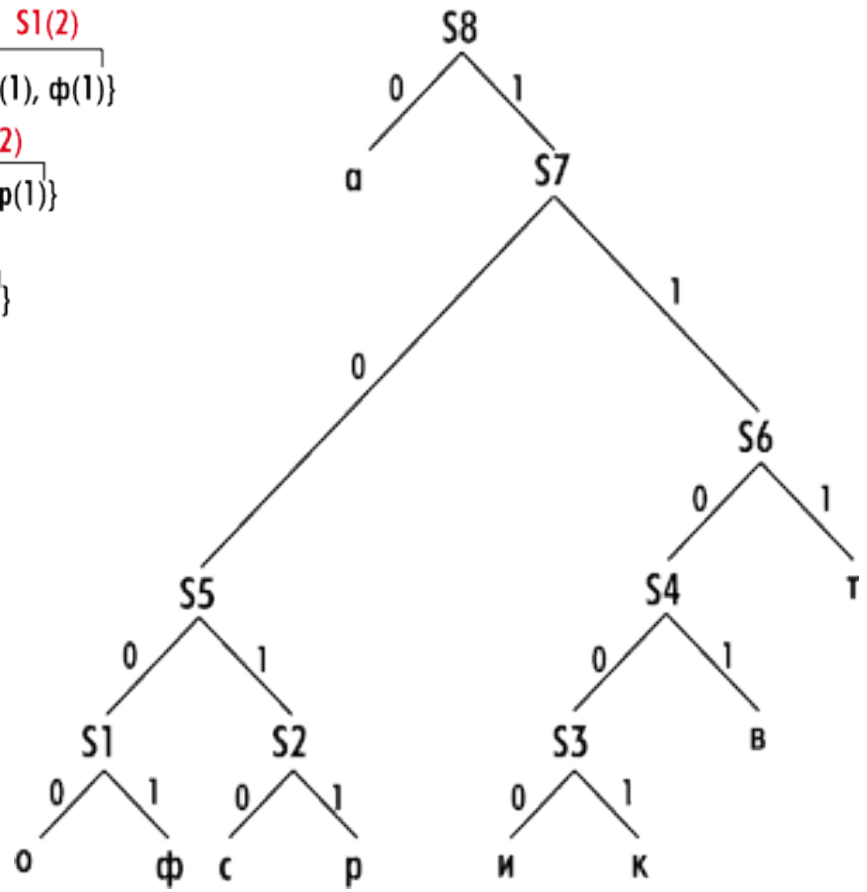
{а(5), S4(3), т(2), S5(4)[S1(2), S2(2)]}

{а(5), S5(4), S6(5)[S4(3), т(2)]}

{а(5), S7(9)[S6(5), S5(4)]}

{S8(14)[S7(9), а(5)]}

{S8(14)}



а 0

т 111

в 1101

и 11000

к 11001

с 1010

р 1011

о 1000

ф 1001

Код Хаффмана имеет минимальную длину среди префиксных.
Не увеличивает размера исходных данных в худшем случае.

Арифметический (интервальный) код

Неалфавитное неразделимое кодирование

$C = c_0 c_1 c_2 ... c_n \to z \in [0, 1);$ $\qquad (0, 1) \underline{\sim} \mathbb{R}$

$I(z) \approx I(C),$ $\qquad$ и чаще всего $I(z) >> 64$ бит $> I(\texttt{double})$

Спасибо за внимание!

МИЭТ
http://miet.ru/

Александра Игоревна Кононова
illinc@mail.ru