Современные проблемы информатики

Кодирование источника. Статистические коды

Фионов Андрей Николаевич

СибГУТИ

2020

Основные понятия

Код – последовательность символов некоторого (кодового) алфавита, сопоставленная буквам или сообщениям источника

Двоичные коды строятся из символов алфавита $\{0, 1\}$

Кодовое слово – последовательность кодовых символов, сопоставленная некоторой букве алфавита источника

Длина кодового слова – количество кодовых символов в слове

Кодовое пространство – множество, из которого выбираются кодовые слова

Виды кодов

Коды *фиксированной длины* – длины всех кодовых слов равны
Коды *переменной длины* – длины (хотя бы некоторых) кодовых слов различны

Разделимые (однозначно декодируемые) коды – коды, позволяющие восстановить (декодировать) символ источника единственным образом
Неразделимые коды – коды, не позволяющие декодировать символ однозначно

Префиксные коды – коды, в которых ни одно кодовое слово не является началом другого кодового слова

Полные коды – коды, в которых кодовые слова полностью покрывают всё кодовое пространство
Неполные коды – коды, оставляющие свободные (незанятые) места в кодовом пространстве

Примеры кодов

$A = \{a, b, c, d, e\}$

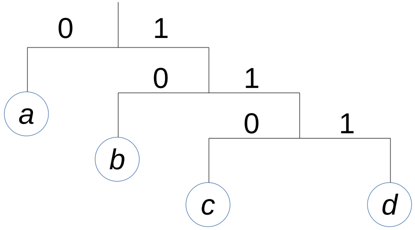
	1	2	3	4
a	000	0	00	1
b	001	1	01	10
c	010	10	100	10000
d	011	11	101	10001
e	100	100	11	10010

Коды фиксированной длины – 1 Коды переменной длины – 2, 3, 4
Разделимые коды – 1, 3, 4 Неразделимые коды – 2
Префиксные коды – 1, 3
Полные коды – 2, 3 Неполные коды – 1, 4

Представление в виде дерева

$A = \{a, b, c, d\}$

a	0
b	10
c	110
d	111

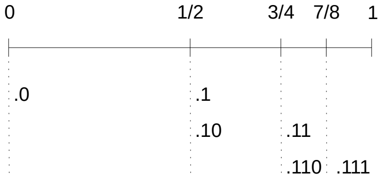


- полный код – всем листьям соответствуют буквы алфавита
- декодирование – проход по дереву

Интервальное представление

$A = \{a, b, c, d\}$

a	0	[0, 1/2)
b	10	[1/2, 3/4)
c	110	[3/4, 7/8)
d	111	[7/8, 1)



- разделимый код – интервалы не перекрываются
- декодирование – попадание в интервал

Неравенство Крафта–Макмиллана

1. Длины кодовых слов $\ell_1, \ell_2, \ldots, \ell_N$ любого двоичного разделимого кода удовлетворяют неравенству

$$\sum_{i=1}^{N} 2^{-\ell_i} \le 1.$$

2. Для любого набора длин, удовлетворяющих этому неравенству, существует префиксный код с такими длинами кодовых слов.

Средняя длина кодового слова

$$\tilde{\ell} = p_1\ell_1 + p_2\ell_2 + \cdots + p_N\ell_N$$

Первая теорема Шеннона

Теорема о кодировании источника.
1. Средняя длина кодового слова разделимого кода не может быть меньше энтропии источника ($\tilde{\ell} \geq H$).
2. Наименьшая средняя длина кодового слова ($\tilde{\ell} = H$) достигается тогда и только тогда, когда каждый символ, появляющийся с вероятностью p, кодируется точно $-\log p$ битами.

Избыточность кода

$$r = \tilde{\ell} - H$$

Код Шеннона. Алфавитный код

Определение. *Кумулятивная вероятность* символа есть сумма вероятностей предшествующих символов

$$q_1 = 0, \quad q_i = \sum_{j=1}^{i-1} p_j, \quad i = 2, \ldots, N$$

$$(q_1 = 0, \quad q_i = q_{i-1} + p_{i-1}, \quad i = 2, \ldots, N)$$

Код Шеннона

Построение.
1. Упорядочиваем буквы алфавита по убыванию вероятностей.
2. Вычисляем кумулятивные вероятности.
3. В качестве кода буквы x берем $\lceil -\log p_x \rceil$ бит дробной части двоичного представления числа q_x.

Свойства кода Шеннона:

- Код является префиксным
- Избыточность $r < 1$ бит на символ

Код Шеннона

Декодируемость (префиксность).
Рассмотрим два кодовых слова для букв a_i и a_j с длинами ℓ_i, ℓ_j, $i < j$.
Вследствие сортировки $\ell_i \le \ell_j$.
Из определения кумулятивных вероятностей $q_j - q_i \ge p_i$.
В соответствии с методом построения кода $\ell_i = \lceil -\log p_i \rceil \ge -\log p_i$.
Отсюда $p_i \ge 2^{-\ell_i}$ и $q_j - q_i \ge 2^{-\ell_i}$.
Значит i-е и j-е кодовые слова отличаются по меньшей мере в одном из ℓ_i разрядов. Код получается префиксным.

Код Шеннона

Избыточность.

$$\lceil -\log p_x \rceil < -\log p_x + 1$$

$$\begin{aligned}
\tilde{\ell} &= \sum_x p_x(\lceil -\log p_x \rceil) \\
&< \sum_x p_x(-\log p_x) + \sum_x p_x \\
&= H(X) + 1
\end{aligned}$$

Код Шеннона

Пример.

$A = \{a, b, c, d, e\}$, $\mathbb{P} = (1/16, 3/16, 1/16, 4/16, 7/16)$, $H = 1.97$

1. $A' = \{e, d, b, a, c\}$, $\mathbb{P} = (7/16, 4/16, 3/16, 1/16, 1/16)$

2. $\mathbb{Q} = (0, 7/16, 11/16, 14/16, 15/16) = (.0000, .0111, .1011, .1110, .1111)$

3. $e \to 00$, $d \to 01$, $b \to 101$, $a \to 1110$, $c \to 1111$

$\tilde{\ell} = 2.4375$, $r = 0.4675$

Алфавитный код

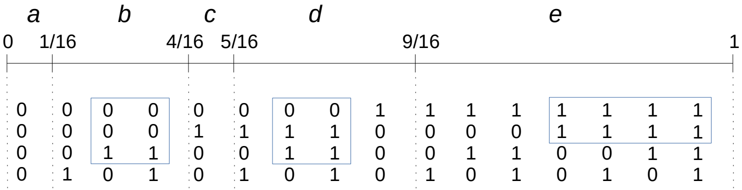
Построение.
1. Вычисляем кумулятивные вероятности.
2. В качестве кода буквы x берем код наибольшего подынтервала, целиком входящего в интервал $[q_x, q_x + p_x)$.
Свойства алфавитного кода:

- Код является префиксным
- Избыточность $r < 2$ бит на символ
- Кодовые слова сохраняют порядок букв алфавита

Алфавитный код

Пример.
$A = \{a, b, c, d, e\}$, $\mathbb{P} = (1/16, 3/16, 1/16, 4/16, 7/16)$, $H = 1.97$

1. $\mathbb{Q} = (0, 1/16, 4/16, 5/16, 9/16) = (.0000, .0001, .0100, .0101, .1001)$



2. $a \to 0000$, $b \to 001$, $c \to 0100$, $d \to 011$, $e \to 11$

$\tilde{\ell} = 2.6875$, $r = 0.7175$

Код Хаффмана (Huffman)

Построение.
1. Находим два символа с наименьшими вероятностями, приписываем одному 0, другому 1.
2. Заменяем в алфавите найденные символы на один новый символ, суммируя их вероятности.

Указанные 2 шага повторяем до тех пор, пока в алфавите не останется один символ.

Код каждого исходного символа получаем, читая приписанные биты в обратном порядке.

Код Хаффмана

Свойства кода Хаффмана:

- Код является префиксным
- Код всегда полный
- Избыточность $r < 1$ бит на символ
- Код является *оптимальным* (самое главное свойство) – для заданного набора вероятностей невозможно построить другой код, имеющий меньшую среднюю длину кодового слова

Код Хаффмана

Пример.
$A = \{a, b, c, d, e\}$, $\mathbb{P} = (1/16, 3/16, 1/16, 4/16, 7/16)$, $H = 1.97$

$a : 0,\ \ c : 1,\ A_1 = \{ac, b, d, e\},\ p(ac) = 2/16$
$ac : 0,\ \ b : 1,\ A_2 = \{acb, d, e\},\ p(acb) = 5/16$
$acb : 0,\ \ d : 1,\ A_3 = \{acbd, e\},\ p(acbd) = 9/16$
$acbd : 0,\ \ e : 1,\ A_4 = \{acbde\},\ p(acbde) = 1$ – конец

$a \to 0000,\ b \to 001,\ c \to 0001,\ d \to 01,\ e \to 1$

$\tilde{\ell} = 2,\ r = 0.03$

Уменьшение избыточности

Единственный способ – кодировать блоки символов.

Пусть m – размер блока. Если рассматривать блоки символов как буквы некоторого “суперaлфавита”, то можно достичь избыточности

$$r < 1/m \text{ бит на символ}$$

Пример.
$A = \{a, b\},\ p_a = 0.2, p_b = 0.8,\ m = 2$

$A^2 = \{aa, ab, ba, bb\},\ p_{aa} = 0.04, p_{ab} = 0.16, p_{ba} = 0.16, p_{bb} = 0.64$

Используем код Хаффмана для A^2
Избыточность на символ исходного алфавита A

$$r < 1/2$$

В общем случае

$$r < 1/m, \quad r \to 0 \text{ при } m \to \infty$$

Сложность блокового кодирования

Код Шеннона, код Хаффмана: нужно делать сортировку символов алфавита

Трудоемкость по времени T, объем памяти M при увеличении размера алфавита N:

$$T = O(N \log N), \quad M = O(N)$$

При кодировании блоков размера m

$$T = O(mN^m \log N), \quad M = O(N^m), \quad m \to \infty$$

Сложность блокового кодирования

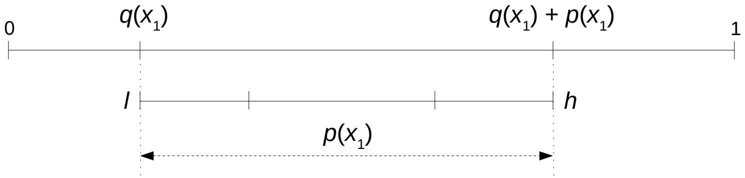
Алфавитный код: сортировку делать не нужно, требуется только вычислять кумулятивные вероятности

При кодировании блоков размера m

$$T = O(m^2), \quad M = O(m), \quad m \to \infty$$

Даёт начало наилучшему методу – арифметическому кодированию

Блоковый алфавитный код



$m = 4$, $X = x_1x_2x_3x_4$

1. $l = q(x_1), \quad h = l + p(x_1)$
2. $l = q(x_1) + p(x_1)q(x_2), \quad h = l + p(x_1)p(x_2)$
3. $l = q(x_1) + p(x_1)q(x_2) + p(x_1)p(x_2)q(x_3),$
$$h = l + p(x_1)p(x_2)p(x_3)$$
4. $l = q(x_1) + p(x_1)q(x_2) + p(x_1)p(x_2)q(x_3) + p(x_1)p(x_2)p(x_3)q(x_4),$
$$h = l + p(x_1)p(x_2)p(x_3)p(x_4)$$

Блоковый алфавитный код

$l = q(x_1) + p(x_1)q(x_2) + p(x_1)p(x_2)q(x_3) + p(x_1)p(x_2)p(x_3)q(x_4)$,
$h = l + p(x_1)p(x_2)p(x_3)p(x_4)$

t – число бит в представлении вероятностей

трудоёмкость умножений: $2tt + 2(2t)t + 2(3t)t = 2t^2(1 + 2 + 3)$

для m символов $2t^2(1 + 2 + \cdots + (m - 1)) = t^2 m(m - 1) = O(m^2)$

Блоковый алфавитный код

Пример.
$p(a) = 1/16, p(b) = 3/16, p(c) = 1/16, p(d) = 4/16, p(e) = 7/16$
$q(a) = 0, q(b) = 1/16, q(c) = 4/16, q(d) = 5/16, q(e) = 9/16$
$X = beed$

1. $l = \frac{1}{16} \quad h = \frac{4}{16}$
2. $l = \frac{1}{16} + \frac{3}{16}\frac{9}{16} = \frac{43}{256} \quad h = \frac{43}{256} + \frac{3}{16}\frac{7}{16} = \frac{64}{256}$

...

4. $l = \frac{14767}{65536} \quad h = \frac{15355}{65536}$

14767 = 0011 1001 1010 1111
15354 = 0011 1011 1111 1010
код = 00111010

Блоковый алфавитный код – метод Рябко

$$l = q(x_1) + p(x_1)q(x_2) + p(x_1)p(x_2)q(x_3) + p(x_1)p(x_2)p(x_3)q(x_4),$$
$$h = l + p(x_1)p(x_2)p(x_3)p(x_4)$$

$$l = q(x_1) + p(x_1)q(x_2) + p(x_1)p(x_2) \cdot (q(x_3) + p(x_3)q(x_4)),$$
$$h = l + (p(x_1)p(x_2)) \cdot (p(x_3)p(x_4))$$

трудоёмкость умножений: $4tt + 2(2t)(2t)$

$T = O(m \log^2 m \log\log m)$ (умножение методом Шёнхаге–Штрассена)

Арифметический код

L – левая граница интервала
R (range) – размер интервала

Вычисление границ интервала

$L \leftarrow 0, R \leftarrow 1;$
FOR $(i = 1, 2, \ldots, m)$ DO
    $L \leftarrow L + Rq(x_i),$
    $R \leftarrow Rp(x_i).$

Основные идеи:

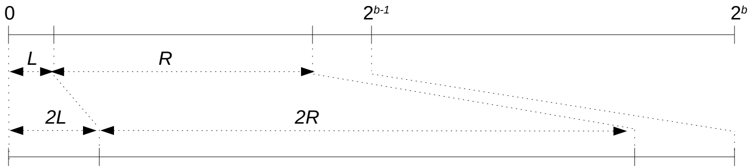
1. Переводим L, R в область целых чисел $([0, 1) \to [0, 2^b))$
2. При умножении на вероятность отсекаем дробную часть (немного жертвуем избыточностью)
3. Масштабируем интервал

Арифметический код – масштабирование

1. Интервал целиком лежит в левой половине

Выдаём на выход кодера бит 0

Масштабируем так:

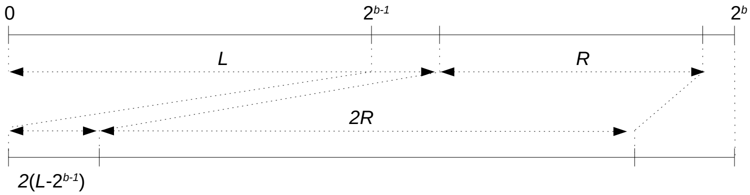


Арифметический код – масштабирование

2. Интервал целиком лежит в правой половине

Выдаём на выход кодера бит 1

Масштабируем так:

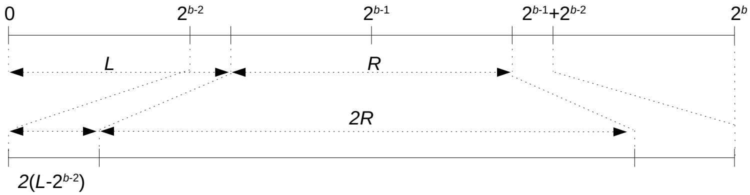


Арифметический код – масштабирование

3. Интервал целиком лежит в пределах 2 и 3 четверти

Увеличиваем на 1 счетчик S.

Масштабируем так:



В будущем при выдаче на выход кодера бита x после него надо будет выдать S противоположных бит

Арифметический код – масштабирование

В результате каждый бит выдаём на выход кодера с помощью следующего алгоритма

$\mathrm{outS}(x)$

выдать бит x,

S раз выдать бит $1 - x$,

$S \leftarrow 0$.

Арифметический код V1

Пусть $p_1, p_2, \ldots, p_N$ – целые, $\sum p_i = d$, т.е. $p(a_i) = p_i/d$.
Тогда $q_1 = 0, q_2, \ldots, q_N, q_{N+1} = d$ тоже целые.
Полный интервал $[0, 2^b)$ – целочисленный.

Кодер: внутреннее состояние $L = 0$, $R = 2^{b-1}$, $S = 0$.

Процедура кодирования Aenc имеет три параметра – левая и правая границы интервала для символа, общий знаменатель вероятностей. Для кодирования сообщения $x_1 x_2 \ldots x_n$ вызываем $\mathrm{Aenc}(q(x_i), q(x_i) + p(x_i), d)$ для $i = 1, 2, \ldots, n$. В ходе своего выполнения Aenc выдаёт биты на выход кодера и формирует новое внутренне состояние. В конце необходимо вывести код подынтервала внутри конечного состояния (максимум 2 бита).

Арифметический код V1

$\mathrm{Aenc}(l, h, d)$

$L \leftarrow L + (R \times l) \operatorname{div} d$ (деление с отсечением дробной части)

$R \leftarrow (R \times (h - l)) \operatorname{div} d$

WHILE $R \leq 2^{b-2}$ DO (масштабирование)

IF $L + R \leq 2^{b-1}$ THEN $\mathrm{outS}(0)$

ELSE IF $L \geq 2^{b-1}$ THEN $\mathrm{outS}(1)$, $L \leftarrow L - 2^{b-1}$

ELSE $S \leftarrow S + 1, L \leftarrow L - 2^{b-2}$

$L \leftarrow 2L, R \leftarrow 2R$

Завершение кодирования

$S \leftarrow S + 1$

IF $L \leq 2^{b-2}$ THEN $\mathrm{outS}(0)$

ELSE $\mathrm{outS}(1)$

Арифметический код V1

Декодер: внутреннее состояние $V =$ первые b бит кода, $L = 0$, $R = 2^{b-1}$.

Декодирование символа

1. $t \leftarrow ((V - L + 1) \times d - 1) \operatorname{div} R$.
2. Находим символ s, для которого $l = q_s \le t < q_{s+1} = h$.
3.

$\mathrm{Adec}(l, h, d)$

$L \leftarrow L + (R \times l) \operatorname{div} d$

$R \leftarrow (R \times (h - l)) \operatorname{div} d$

WHILE $R \le 2^{b-2}$ DO

IF $L + R \le 2^{b-1}$ THEN ничего не делать

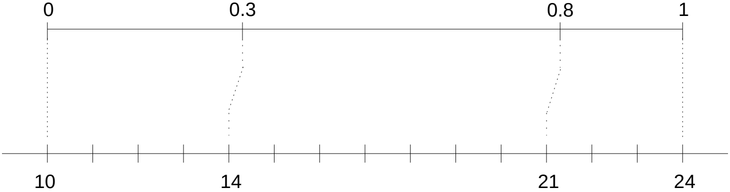
ELSE IF $L \ge 2^{b-1}$ THEN $L \leftarrow L - 2^{b-1}$, $V \leftarrow V - 2^{b-1}$

ELSE $L \leftarrow L - 2^{b-2}$, $V \leftarrow V - 2^{b-2}$

$L \leftarrow 2L, R \leftarrow 2R$, $V \leftarrow 2V+$ очередной бит кода

Арифметический код V1 – избыточность

Пример. $A = \{a, b, c\}, p_a = 0.3, p_b = 0.5, p_c = 0.2$



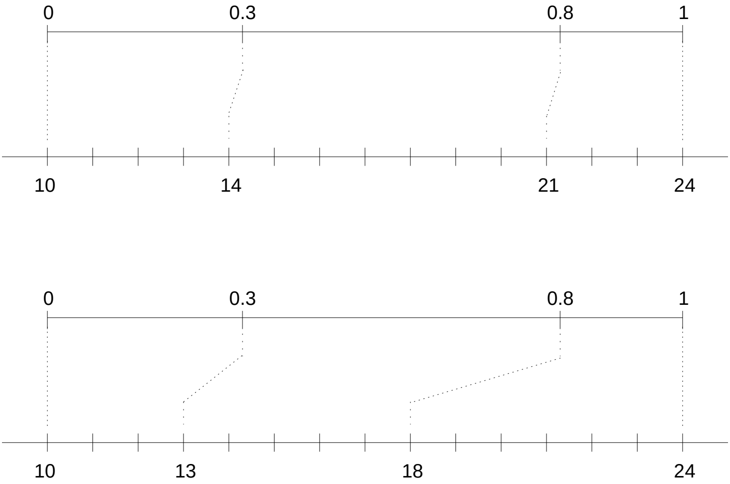
$$
\begin{array}{ll}
I(a) = -\log 0.3 = 1.74 & \ell_a = -\log(4/14) = 1.81 \\
I(b) = -\log 0.5 = 1.00 & \ell_b = -\log(7/14) = 1.00 \\
I(c) = -\log 0.2 = 2.32 & \ell_c = -\log(3/14) = 2.22
\end{array}
$$

$$H = 1.4855 \quad \tilde{\ell} = 1.4867 \quad r = 0.0012$$

Арифметический код V2

Идея.

$$(R \times l) \operatorname{div} d \quad \to \quad (R \operatorname{div} d) \times l$$



Арифметический код V2

Кодер: внутреннее состояние $L = 0$, $R = 2^{b-1}$, $S = 0$.

$\mathrm{Aenc2}(l, h, d)$

$k \leftarrow R \operatorname{div} d$

$L \leftarrow L + k \times l$

IF $h < d$ THEN $R \leftarrow k \times (h - l)$

ELSE $R \leftarrow R - k \times l$

масштабирование

Арифметический код V2

Декодер: внутреннее состояние $D =$ первые b бит кода, $R = 2^{b-1}$ $(D = V - L)$.

Декодирование символа
1. $k \leftarrow R \operatorname{div} d$, $t \leftarrow \min(d-1, D \operatorname{div} k)$.
2. Находим символ s, для которого $l = q_s \leq t < q_{s+1} = h$.
3.

$\mathrm{Adec2}(l, h, d)$

$$D \leftarrow D - k \times l$$

IF $h < d$ THEN $R \leftarrow k \times (h - l)$

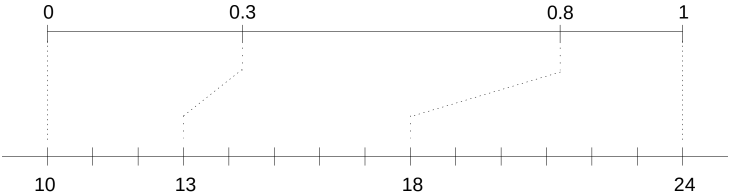
ELSE $R \leftarrow R - k \times l$

WHILE $R \leq 2^{b-2}$ DO

$R \leftarrow 2R$, $D \leftarrow 2D+$ очередной бит кода

Арифметический код V2 – избыточность

Пример. $A = \{a, b, c\}$, $p_a = 0.3$, $p_b = 0.5$, $p_c = 0.2$



$$I(a) = -\log 0.3 = 1.74 \quad \ell_a = -\log(3/14) = 2.22$$
$$I(b) = -\log 0.5 = 1.00 \quad \ell_b = -\log(5/14) = 1.49$$
$$I(c) = -\log 0.2 = 2.32 \quad \ell_c = -\log(6/14) = 1.22$$

$$H = 1.4855 \quad \tilde{\ell} = 1.6539 \quad r = 0.1684$$

Арифметический код – избыточность

$$r_{\mathrm{ac1}} < N(\tau + \log \mathrm{e}) \cdot 2^{-(b-2)} + 2/n$$

$$r_{\mathrm{ac2}} < \log \mathrm{e} \cdot 2^{-(b-2)+\tau} + 2/n$$

N – размер алфавита
τ – количество бит в представлении вероятности
b – битовый размер регистров кодера
n – длина сообщения

Кодирование источника. Статистические коды

К О Н Е Ц