УДК 519.72

Обзор методов неискажающего кодирования дискретных источников *

В. Н. Потапов

Аннотация

В обзоре рассмотрены основные задачи и конструкции теории неискажающего кодирования дискретных источников: побуквенное, адаптивное и универсальное кодирование, принцип кратчайшего описания, построение дерева контекстов и преобразование Барроуза–Уилера. Описаны наиболее известные методы сжатия данных: блочное, равномерное по выходу и арифметическое кодирование, схема кодирования Лемпела–Зива и методы интервального кодирования. Приведены оценки избыточности и трудоемкости перечисленных методов. Даны схемы доказательства для некоторых наиболее важных утверждений. Кроме того, рассмотрены задачи рандомизации сообщений и кодирования с синхронизацией, а также способы кодирования текстов на естественных языках и источников с низкой энтропией.

Введение

Теория кодирования дискретных источников исследует задачу сжатия сообщений без потери информации. Под сообщением подразумевается конечное или бесконечное слово в некотором алфавите, порожденное стационарным источником. Источником называется последовательность одинаково распределенных случайных величин, множество значений которых — алфавит источника. Кодированием называется инъективное отображение множества слов алфавита источника в множество двоичных слов, а стоимостью кодирования — отношение средней длины кода сообщения к длине сообщения. В статье [100], положившей начало современной теории информации в целом и теории кодирования источников в частности, К. Шеннон показал, что нижней гранью стоимости кодирования является энтропия источника. Избыточность — разность между стоимостью и энтропией — является основным критерием качества кодирования. Другими важными характеристиками кодирования являются объем памяти, который требуется при программной реализации метода, и среднее время кодирования и декодирования, измеряемое в операциях над битами. Основной областью применения алгоритмов кодирования являются компьютерные программы сжатия данных (архиваторы). Методы кодирования могут быть использованы также в криптографии (см. [23, 30, 29, 62, 65]), задачах прогнозирования (см. [20, 79]) и поиска информации (см. [11]).

*) Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (код проекта 99-01-00531) и Федеральной целевой программы "Интеграция" (проект 1997 года №473)

В обзоре сделана попытка проследить основные направления развития теории кодирования дискретных источников от задачи оптимального кодирования известного источника к задаче оптимального универсального кодирования семейства источников и задаче построения модели источника по сообщению; а также от побуквенного кодирования к арифметическому кодированию и схеме кодирования Лемпела–Зива. Кроме того, рассмотрены самосихронизирующиеся и омофонные коды, а также способы кодирования двух часто встречающихся на практике видов источников: текстов на естественных языках и источников с низкой энтропией.

Основная часть, приведенных в обзоре сведений содержится в монографии Р. Е. Кричевского [11], обзорных статьях А. Д. Винера с соавторами [117], Н. Мерхева и М. Федера [79], А. Баррона с соавторами [40], а также кандидатских диссертациях А. В. Кадача [5], А. Н. Фионова [30], М. П. Шаровой [35].

1. Источники сообщений

Пусть $A = \{a_1, ..., a_k\}$ — конечный алфавит и $x \in A^\infty$. Будем обозначать через x_i^j подслово последовательности x, начиная с i-ой и заканчивая j-ой буквой, а через x^n начало последовательности x длины n, т. е. $x_i^j = x_i x_{i+1} \dots x_j$ и $x^n = x_1 x_2 \dots x_n$. *Дискретным источником* X называется дискретный случайный процесс со значениями в A. Источник полностью задается вероятностями $Pr(X^n = x^n)$, которые определены для всех $x^n \in A^n$ и целых $n > 0$ и удовлетворяют равенствам $\sum_{i=1}^k Pr(X^{n+1} = x^n a_i) = Pr(X^n = x^n) \geq 0$ и $\sum_{x^n \in A^n} Pr(X^n = x^n) = 1$. Тогда $Pr(X_i^j = x_i^j) = \sum_{y_i^j = x_i^j} Pr(X^j = y^j)$. Источник X называется *стационарным*, если для всех целых $t > 0$ и $x_i^j \in A^{j-i+1}$ справедливы равенства

$$Pr(X_i^j = x_i^j) = Pr(X_{i+t}^{j+t} = x_i^j).$$

Введем обозначение $P(x^n) = Pr(X^n = x^n)$. Для каждого стационарного источника X равенство

$$H(X) = \lim_{n \to \infty} \frac{1}{n} E(-\log P(X^n)) \tag{1}$$

определяет неотрицательную величину $H(X)$ (см., например, [3, 11]), которая называется *энтропией* источника.

Основным видом стационарных дискретных источников, изучаемых в теории кодирования, являются марковские источники с конечным числом состояний. Пусть S — множество состояний источника X с алфавитом A. Тогда для каждой пары $\sigma \in S$ и $a \in A$ определена вероятность $P(a|\sigma) \geq 0$ порождения источником буквы a в состоянии σ и справедливы равенства $\sum_{i=1}^k P(a|\sigma) = 1$. Кроме того, задана функция $\mu : S \times A \to S$, определяющая состояние $\sigma' = \mu(\sigma, a)$, в которое переходит источник после порождения буквы a в состоянии σ. Ясно, что последовательность состояний $\sigma_0 \sigma_1 \sigma_2 \dots$ марковского источника представляет собой марковскую цепь. В дальнейшем будем полагать, что эта цепь является неразложимой и непереодической. Для полного определения источника X нужно задать начальное состояние $\sigma_0 \in S$. Тогда $Pr(X^n = x^n) = \prod_{i=0}^{n-1} P(x_{i+1}|\sigma_i)$, где $\sigma_{i+1} = \mu(\sigma_i, x_{i+1})$. Таким образом, *марковский источник* X задается алфавитом A, множеством состояний S, функцией μ и распределениями вероятностей $P(a_i|\sigma)$ в каждом состоянии, что будем кратко выражать равенством $X = \langle A, S, \mu, P \rangle$.

Обозначим через $q_{\sigma\tau}$ вероятности перехода источника $X = \langle A, S, \mu, P \rangle$ из состояния σ в состояние τ, т. е. $q_{\sigma\tau} = \sum_{\sigma:\mu(\sigma,a_i)=\tau} P(a_i|\sigma)$. Тогда стационарные вероятности

q_σ марковской цепи $\sigma_0\sigma_1\sigma_2\ldots$ можно получить из системы уравнений $q = qQ$ и $\sum_{\sigma\in S} q_\sigma = 1$, где Q — матрица переходных вероятностей $\{q_{\sigma\tau}\}$. Известно (см., например, [3]), что для марковского источника $X = \langle A, S, \mu, P\rangle$ справедливо равенство

$$H(X) = \sum_{\sigma\in S} q_\sigma H_\sigma, \tag{2}$$

где

$$H_\sigma = -\sum_{i=1}^{k} P(a_i|\sigma)\log P(a_i|\sigma).$$

Марковский источник $X = \langle A, S, \mu, P\rangle$ с единственным состоянием называется *источником без памяти* или *источником Бернулли*. Для него справедлива предложенная К. Шенноном [100] формула

$$H(X) = -\sum_{i=1}^{k} P(a_i)\log P(a_i). \tag{3}$$

Стационарный источник называется *марковским r-го порядка*, если вероятность появления очередой буквы зависит только от r предыдущих букв, т. е. множество состояний S источника можно отождествить с A^r.

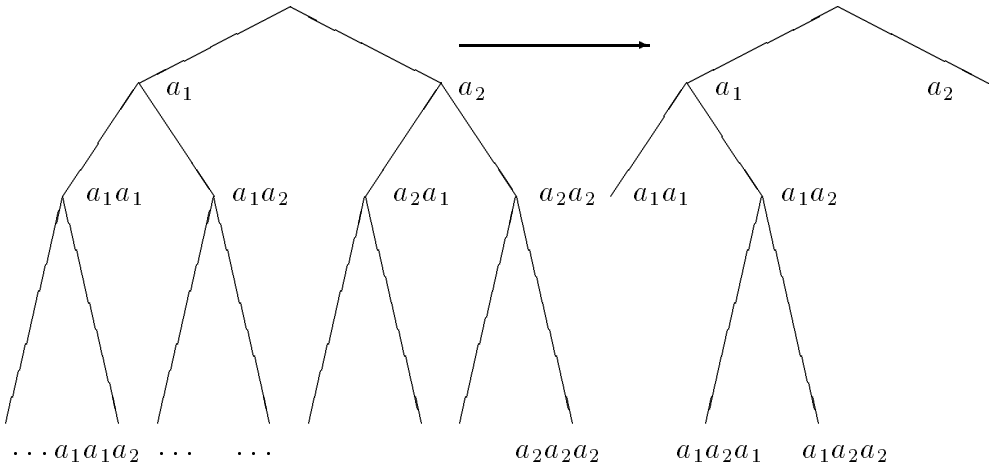

Рис. 1.

Множество состояний марковского источника r-го порядка представим в виде листьев дерева. Пример такого дерева для двоичного алфавита и $r = 3$ изображен на рис. 1 (слева). Состояния σ и σ' будем называть эквивалентными, если $P(a_i|\sigma) = P(a_i|\sigma')$ для всех $a_i \in A$ и листья, соответствующие состояниям σ и σ', являются братьями. Объединяя каждую пару эквивалентных состояний в новое состояние, получим минимальное дерево состояний S', не содержащее эквивалентных состояний. Пример преобразования приведен на рис. 1. Полученный источник-дерево $\langle A, S', \mu', P\rangle$ может оказаться не марковским как, например, на рис. 1 (после порождения буквы a_1 в состоянии $\sigma = a_2$ следующее состояние неопределено). Однако каждый источник-дерево очевидно может быть дополнен до марковского обратной процедурой. Этот вид источников был введен Й. Риссаненом [90] и будет рассмотрен в разделе 10.

А. Н. Колмогоровым [6] и В. Д. Гоппой [4] показано, что многие задачи теории информации могут быть рассмотрены не только в вероятностной, но и в комбинаторной

постановке. Соответственно могут быть рассмотрены не только вероятностные (как это сделано выше), но и комбинаторные источники сообщений. В частности, подробное исследование комбинаторных регулярных (аналог марковских) источников содержится в книге А. А. Маркова [14]. Кодирование комбинаторных источников редких событий рассмотрено в работах Р. Е. Кричевского и автора [69, 15]. Универсальное кодирование семейства комбинаторных источников предложено Б. Я. Рябко [17].

2. Основные определения теории кодирования источников

Пусть $E = \{0, 1\}$, обозначим через $E^* = \cup_{n=1}^{\infty} E^n$ множество всех двоичных слов. Инъективное отображение $f : A \to E^*$ называется *побуквенным дешифруемым кодированием*, если произвольный конечный набор кодовых слов $f(a_{i_1}) \dots f(a_{i_n})$, записанных слитно, однозначно разделяется на кодовые слова. Известно несколько алгоритмов, позволяющих определить является ли отображение дешифруемым кодированием (см., например, [12, 48, 93]).

Множество $M \subset E^*$ двоичных слов называется *префиксным*, если никакое слово $v \in M$ не является префиксом (началом) другого слова $u \in M$. Инъективное отображение $f : A \to E^*$ называется *побуквенным префиксным кодированием*, если множество $f(A)$ кодовых слов — префиксное. Нетрудно заметить, что каждому префиксному множеству соответствует множество листовых вершин некоторого двоичного дерева. Непосредственно из этого можно заключить, что для произвольного побуквенного префиксного кодирования f справедливо неравенство Крафта

$$\sum_{i=1}^{k} 2^{-l_i} \le 1, \tag{4}$$

где $l_i = |f(a_i)|$ — длина двоичного слова $f(a_i)$. Верно и обратное: для любого набора целых чисел $l_i > 0$, удовлетворяющего неравенству (4), найдется префиксное кодирование f такое, что $l_i = |f(a_i)|$. Префиксный код называют *полным*, если в формуле (4) имеет место равенство. Ясно, что произвольное префиксное кодирование является дешифруемым, а обратное, вообще говоря, неверно. Однако, Б. Макмилланом [78] доказано неравенство (4) для произвольного побуквенного дешифруемого кодирования.

Блочное кодирование $f : A^n \to E^*$ можно свести к побуквенному, определив новый алфавит $B = A^n$. Вообще, *кодированием* называется инъективное отображение $f : A^* \to E^*$, где $A^* = \cup_{n=1}^{\infty} A^n$ — множество слов алфавита A. Побуквенное дешифруемое кодирование $f : A \to E^*$ является частным случаем кодирования, поскольку допускает доопределение $f(x^n) = f(x_1) \dots f(x_n)$. Кодирование $f : A^* \to E^*$ будем называть *префиксным*, если $f(A^n)$ — префиксное множество для всех целых $n > 0$.

Стоимостью $C(f, X)$ кодирования f источника X называется среднее число битов, которое нужно затратить для кодирования одной буквы исходного сообщения, т. е.

$$C(f, X) = \limsup_{n \to \infty} \frac{1}{n} L_n(f, X), \quad \text{где} \quad L_n(f, X) = E|f(X^n)|. \tag{5}$$

В частности, если X — источник без памяти, а f — побуквенное кодирование, то

$$C(f, X) = L_1(f, X) = \sum_{i=1}^{k} P(a_i)|f(a_i)|. \tag{6}$$

Избыточностью кодирования f слова $x^n \in A^n$ на букву исходного сообщения называется величина $\frac{1}{n}(|f(x^n)| - \log \frac{1}{P(x^n)})$. *Средней избыточностью* кодирования (на букву исходного сообщения) назвается разность между стоимостью кодирования и энтропией источника (см. (1)), т. е.

$$R(f, X) = C(f, X) - H(X) = \limsup_{n\to\infty} R_n(f, X), \tag{7}$$

где $R_n(f, X) = \frac{1}{n}E(|f(X^n)| - \log \frac{1}{P(X^n)})$.
Аргументы функций C, R, L_n, R_n будем в дальнейшем опускать в тех случаях, когда они ясны из контекста.

Если f — префиксное кодирование, то из неравенства (4) и выпуклости вверх функции $\log t$ следует неравенство $R_n(f, X) \geq 0$. Для произвольного кодирования f Леун-Ян-Ченгом и Т. Ковером [72] доказано неравенство

$$R_n(f, X) \geq -\frac{\log n + \log\log k + c}{n},$$

где $c > 0$ — константа, независящая от k и n. Таким образом средняя избыточность R произвольного кодирования неотрицательна.

3. Побуквенное и блочное кодирование

Сначала рассмотрим побуквенное префиксное кодирование источников без памяти. Задача построения для произвольного источника без памяти оптимального, т. е. имеющего наименьшую стоимость, побуквенного префиксного кодирования была решена Д. Хаффменом [64]. Он предложил следующий индуктивный по размеру алфавита алгоритм построения оптимального кодирования. Сначала упорядочиваем буквы алфавита по убыванию их вероятностей. Затем объединяем две наименее вероятные буквы a_{k-1} и a_k в одну новую букву a', вероятность которой определяем как сумму вероятностей букв a_{k-1} и a_k. Если код Хаффмена для нового алфавита, содержащего на одну букву меньше, известен, то полагаем коды неизмененных букв прежними, а коды букв a_{k-1} и a_k определяем как код буквы a' с добавлением в конце 0 или 1 соответственно.

Широко известны побуквенные префиксные коды Шеннона-Фано и Шеннона [100]. Первый из них строится по индукции. Сначала упорядочиваем буквы алфавита по убыванию их вероятностей. Затем разделим алфавит на две части, имеющие наиболее близкие вероятности, не меняя порядка букв. Запишем 0 в качестве первого символа кода для всех букв первой половины алфавита и 1 в качестве первого символа кода для всех букв второй половины. Каждую из двух половин алфавита делим опять на две части с возможно близкими вероятностями и приписываем к коду первых частей 0, к коду вторых частей — 1. Процесс деления продолжаем пока в каждой из частей не останется лишь по одной букве.

Алгоритм построения кода Шеннона позволяет получить код произвольной буквы независимо от других с помощью только арифметических операций. Пусть буквы упорядочены по убыванию их вероятностей. Определим числа $\delta_1 = 0$ и $\delta_{i+1} = \delta_i + P(a_i)$. В качестве кодового слова $f(a_i)$ возьмем первые после запятой $\lceil -\log P(a_i) \rceil$ двоичных знаков числа δ_i. Очевидно, что избыточность кодирования R каждой буквы не превышает 1 и из формул (3) и (6) получается оценка средней избыточности кодирования Шеннона источника без памяти

$$R = C - H = \sum_{i=1}^{k} P(a_i)(\lceil -\log P(a_i) \rceil + \log P(a_i)) \leq 1. \tag{8}$$

Можно заметить, что избыточность кодирования Шеннона не меньше избыточности кодирования Шеннона–Фано, которая в свою очередь не меньше избыточности оптимального кодирования Хаффмена (в среднем). Достаточно рассмотреть источники с двухбуквенным алфавитом и вероятностями букв 0, 1 и 1/2, 1/2, чтобы убедиться, что верхняя и нижняя оценка избыточности 1 и 0 соответственно достигаются для всех трех рассмотренных кодов.

Однако можно получить более точные оценки избыточности кодирования, если известны вероятности букв. Нетривиальные нижние оценки избыточности побуквенного префиксного кодирования были получены Р. Е. Кричевским и Г. Л. Ходаком [7, 11]. Различные верхние оценки избыточности кодирования Хаффмена были получены в работах Р. Галлагера [58], Р. Капоселли и А. Ди Санти [46], Д. Манстетина [77]. Последний, в частности, доказал неравенство

$$R(f_0, X) \le \log\left(\frac{2\log e}{e}\right) + p\frac{2\log e}{e},$$

где f_0 — кодирование Хаффмена, p — максимальная вероятность буквы, порожденной источником без памяти X и e — постоянная Эйлера.

Объем памяти $V(f)$, требуемый для кодирования и декодирования побуквенного префиксного кода, почти линейно зависит от длины алфавита, а время кодирования $T(f)$ на букву исходного сообщения прямо пропорционально стоимости кодирования.

Побуквенное префиксное кодирование можно использовать для кодирования блоков из n букв исходного алфавита. Пользуясь формулами (7) и (8), можно доказать теорему кодирования Шеннона [100] (см. также [3, 11]) для нетеряющего информацию кодирования:
1) избыточность кодирования неотрицательна,
2) для произвольного стационарного источника и $\varepsilon > 0$ найдется блочное префиксное кодирование, избыточность которого не превышает ε.

Практическое применение блочного кодирования ограничивает быстрый рост объема используемой памяти при увеличении длины блока. Для кодирования f_n блоками длины n объем памяти $V(f_n) = O(k^n)$ растет экспоненциально, в то время как избыточность убывает не быстрее чем линейно $R_n(f_n, X) \ge \frac{c}{n}$ при $n \to \infty$, где $c > 0$ — некоторая константа (см.[7]).

Марковский источник с множеством состояний S как и источник Бернулли можно кодировать побуквенным кодом. Для этого нужно разделить сообщение на $|S|$ подпоследовательностей, состоящих из букв, порожденных в каком-то одном состоянии, и затем кодировать каждую подпоследовательность собственным префиксным кодом, учитывающим вероятности букв. В этом случае средняя избыточность кодирования с использованием любого из трех описанных кодов не превышает 1 (это следует из формул (2) и (8)), а объем требуемой памяти линейно зависит от $k|S|$.

Обычно задача построения префиксного побуквенного кода ставится для источника с конечным алфавитом, но проблема кодирования сообщений источника с счетным алфавитом также возникает весьма часто. Обычная двоичная запись натуральных чисел не является префиксным кодом. Простейшим способом ее преобразования в префиксный код является добавление перед двоичной записью числа блока из нулей равного по длине двоичной записи. Естественно этот двоичный код увеличивает длину двоичной записи вдвое. Эффективный полный префиксный код натурального ряда предложил В. И. Левенштейн [13]. Длина кодового слова Левенштейна числа

n удовлетворяет асимптотическому равенству

$$l(n) = \log n + \log\log n(1 + o(1)) \tag{9}$$

при $n \to \infty$. Впоследствии подобные коды были предложены П. Элайесом [54], К. Стоутом [103], а также С. Эвеном и М. Роде [56]. Из работы Р. Альсведе с соавторами [38] следует, что для длины произвольного префиксного кода натурального ряда выполнено (при $n \to \infty$) неравенство

$$l(n) \geq \log n + \log\log n - c, \tag{10}$$

где $c > 0$ — константа, независящая от кода.

4. Омофонное кодирование и кодирование с синхронизацией

Для решения некоторых специальных задач на метод кодирования помимо минимизации избыточности и сложности вычислений налагают дополнительные требования. В частности, с целью защиты информации от незаконного доступа рассматривается задача рандомизации сообщений: необходимо закодировать сообщение так, чтобы символы кодовой последовательности 0 и 1 были независимы и равновероятны.

В первом приближении эта задача решается с помощью произвольного кодирования с низкой избыточностью, поскольку вероятности символов 0 и 1 в кодовой последовательности неизбежно стремятся к $1/2$ при стремлении к нулю избыточности кодирования. Однако, существуют специальные методы решения задачи рандомизации сообщений. К. Гюнтер [62] предложил использовать для этой цели омофонное кодирование. В отличие от обычного при омофонном кодировании одному сообщению могут соответствовать различные кодовые слова. В этом случае дешифруемость уже не является необходимым условием правильного декодирования омофонного кода, если в различных разбиениях сообщения на кодовые слова на одинаковых местах находятся омофоны — коды одной и той же буквы. Свойства такого кодирования рассмотрены, например, А. Вебером и Т. Хедом [108]. Для решения задачи рандомизации сообщений можно ограничиться префиксным омофонным кодированием. Например, для рандомизации источника Бернулли с вероятностями букв $P(a_1) = 3/4$, $P(a_2) = 1/4$ можно использовать омофонное кодирование f: $Pr(f(a_1) = 0) = 2/3$, $Pr(f(a_1) = 10) = 1/3$, $Pr(f(a_2) = 11) = 1$. Кодовые слова 0 и 01 — омофоны.

Для выбора омофона требуется решить задачу генерации случайной величины с заданным распределением. Поэтому важной характеристикой омофонного кодирования является число случайных бит, используемых при кодировании одного символа. В работе Дж. Мэсси с соавторами [65] было предложено и исследовано оптимальное (т. е. имеющее минимальную избыточность) омофонное префиксное побуквенное кодирование. Установлены следующие оценки его эффективности: $R < 2$, $\eta < 4$, где η — число случайных бит на символ сообщения. Задача построения омофонного кодирования источника без памяти с произвольно малыми избыточностью и числом случайных бит решена Б. Я. Рябко и А. Н. Фионовым [23, 29, 30] на основе арифметического кодирования (см. раздел 6).

Разработка методов помехоустойчивого кодирования является одной из центральных задач теории информации. Она обычно решается за счет введения дополнительной избыточности с помощью кодов, исправляющих ошибки. Однако задачи

локализации ошибок можно решать и без введения проверочных символов, например, с помощью кодирования с синхронизацией.

Слово $v \in E^*$ называется *синхронизатором* кода $f(A)$, если для суффикса u произвольного кодового слова конкатенация uv является кодовой последовательностью, т. е. $uv \in f(A^*)$. Если v — синхронизатор кода $f(A)$, то после любой ошибки декодирование будет выполняться неправильно не дольше, чем встретится синхронизатор. Например, для кода $f(A) = \{1, 01, 00\}$ синхронизатором является слово 1. Если в сообщении 00,1,01,1,01 произошла ошибка в первом разряде, то сообщение 1,01,01,1,01 будет декодироваться правильно начиная с синхронизатора, т. е. после третьей буквы. Необходимыми условиями существования синхронизатора префиксного кода являются, во-первых, выполнение равенства Крафта (4) для набора длин кодовых слов $l_1, \ldots, l_k$, а, во-вторых, равенство НОД$(l_1, \ldots, l_k) = 1$.

М. Шютценберже [99] (см. также [12, 14]) показал, что эти два условия являются достаточными для существования префиксного кода с синхронизатором, у которого длины кодовых слов равны $l_1, \ldots, l_k$. Т. Фергюсон и Дж. Рабинович [57] предложили рассматривать коды содержащие синхронизатор как кодовое слово. Такие коды называют *самосинхронизирующимися*. В [57] предложены достаточные условия существования у источника Бернулли самосинхронизирующегося кода Хаффмена. Б. Монтгомери и Дж. Абрахамс [82], а также Р. Капоселли с соавторами [47] предложили алгоритмы построения близких к оптимальным самосинхронизирующихся кодов. Синхронизатор кода позволяет локализовать ошибку, но не всегда дает возможность определить количество неправильно декодированных букв, т. е. после декодирования последовательность букв может оказаться сдвинутой. Последнего недостатка лишены сильно самосинхронизирующиеся коды, предложенные В.-М. Ламом и С. Кулкарни [70].

Не менее важным является исследование средней задержки синхронизации. Говорят, что синхронизация имеет *задержку* τ, если после того как в позиции t произошла ошибка, только начиная с позиции $\tau + t$ последовательность декодируется правильно. М. Титченер [104] предложил метод построения кодов с небольшой средней задержкой синхронизации.

5. Равномерное по выходу кодирование

Равномерное по выходу кодирование также позволяет локализовать ошибки без введения дополнительной избыточности. Ошибка в одном кодовом слове не может повлиять на декодирование других кодовых слов, если все кодовые слова имеют одинаковую длину. Ясно, что побуквенное или блочное равномерное по выходу кодирование не может сжимать данные. Основная идея равномерного по выходу кодирования состоит в том, что разные по длине, но близкие по вероятности слова алфавита источника кодируются всеми возможными блоками из нулей и единиц одинаковой длины. Чтобы любую последовательность букв входного алфавита можно было закодировать, слова в k-буквенном алфавите A, которым сопоставляются кодовые слова, должны соответствовать листьям некоторого полного k-ичного дерева. Например, дерево на рис. 1 (справа) порождает равномерный по выходу код: $f(a_1a_1) = 00$, $f(a_1a_2a_1) = 01$, $f(a_1a_2a_2) = 10$, $f(a_2) = 11$.

Оптимальным равномерным по выходу кодированием называется кодирование, имеющее минимальную среднюю избыточность среди кодов с определенной длиной кодовых слов. Способ построения оптимального равномерного кодирования для источников Бернулли был независимо предложен Г. Л. Ходаком [32] и Б. П. Танстэлом

[106]. Дерево, соответствующее оптимальному равномерному по выходу кодированию с длиной m кодовых слов, можно построить по индукции, начиная с дерева, состоящего только из корня. Пусть имеется некоторое дерево $\Delta \subset A^*$. Выберем лист $x \in \Delta$ с наибольшей вероятностью и добавим к дереву всех сыновей. Эту процедуру будем повторять пока число листьев дерева не превышает 2^m. Листья $x \in A^*$ получившегося дерева будем кодировать различными двоичными словами длины m.

Ч. Чокенс и Ф. Виллемс [105] показали, что существует равномерное по выходу кодирование произвольного марковского источника со сколь угодно малой избыточностью. С. Савари и Р. Галлагер [97] показали, что кодирование Танстэла–Ходака является асимптотически (при $m \to \infty$) близким к оптимальному для марковских источников, причем его средняя избыточность убывает как c/m, где константа $c > 0$ зависит только от источника и точно определена в работе [97].

6. Арифметическое кодирование

История разработки и исследования одного из наиболее популярных и эффективных методов сжатия данных — арифметического кодирования — восходит к классическому коду Шеннона. Э. Н. Гильберт и Э. Ф. Мур [60] предложили блочный код, подобный коду Шеннона, но не требующий предварительного упорядочивания блоков из n букв по вероятностям. А именно, пусть X — стационарный источник в алфавите A. Все наборы из n букв алфавита A упорядочим лексикографически, т. е. $a_{i_1}, a_{i_2}, \ldots, a_{i_n} \prec a_{j_1}, a_{j_2}, \ldots, a_{j_n}$, если $i_m = j_m$ при $m < l$ и $i_l < j_l$. Для каждого слова $x \in A^n$ определим величину $Q(x) = \sum\limits_{y \prec x, y \in A^n} P(y)$. В качестве кода $f(x^n)$ рассмотрим $\lceil -\log P(x^n) \rceil + 1$ двоичных знаков после запятой числа $Q(x^n) + P(x^n)/2$. Полученное кодирование является префиксным и из формулы (7) следует, что $R_n \leq 2/n$.

Процесс кодирования можно представить как разделение отрезка $[0, 1]$ на непересекающиеся полуинтервалы $[Q(x^n), Q(x^n) + P(x^n))$. Кодовое слово $f(x^n)$ оказывается числителем двоично рационального числа со знаменателем $2^{\lceil -\log P(x^n) \rceil + 1}$, попавшего в полуинтервал $[Q(x^n), Q(x^n) + P(x^n))$. П. Элайесом (см. [36, 112]) была предложена процедура последовательного вычисления границ полуинтервала $[Q(x^n), Q(x^n) + P(x^n))$ по мере поступления букв блока x^n. Она естественно вытекает из того, что соответствующий слову $x \in A^*$ полуинтервал по определению разделяется на полуинтервалы, соответствующие словам $xa_1, xa_2, \ldots, xa_k$. Однако в таком виде арифметическое кодирование невозможно использовать на практике при больших n, поскольку требуемая точность вычислений быстро возрастает при увеличении длины блока. Й. Риссанен [86, 87] предложил алгоритм округленного вычисления границ полуинтервала, использующий арифметические операции с числами ограниченной длины. Этот алгоритм позволяет кодировать слова $x^n \in A^n$ с любой наперед заданной избыточностью при $n \to \infty$. Кроме того, в методе Й. Риссанена кодовое слово $f(x^n)$ можно вычислять и передавать поразрядно по мере поступления букв слова x^n. Это усовершенствование оказалось решающим на пути практического использования арифметического кодирования.

Существует несколько версий арифметического кодирования Й. Риссанена, наиболее известная из них разработана И. Уиттеном с соавторами [114]. Оценки эффективности этого алгоритма для источников без памяти содержатся в работе Б. Я. Рябко и А. Н. Фионова [24]:

$$R_n \leq \frac{k(\tau + \log e)}{2^{t-2}} + \frac{2}{n}, \ V = O(t), \ T = O(t \log t \log\log t),$$

где k — объем алфавита, τ — количество битов в представлении вероятностей букв и $t > \tau + 2$ — точность арифметических операций в битах.

Отличные от метода Й. Риссанена варианты арифметического кодирования предложены Ю. М. Штарьковым [36] и Б. Я. Рябко [21, 94].

7. Универсальное кодирование

Чтобы декодировать сообщение нужно знать не только кодовую последовательность, но и функцию, с помощью которой сообщение закодировано, или метод кодирования должен быть универсальным, т. е. пригодным для эффективного кодирования всех источников из некоторого множества.

Рассмотрим некоторое параметрическое семейство M источников, где источник с параметром $\theta \in \Theta$ определяется свом распределением $P_\theta(x)$ вероятностей порождения сообщений $x \in A^*$. Гладкое (в смысле зависимости P_θ от θ) параметрическое семейство источников будем называть *моделью*. Целое число $d > 0$ называется *размерностью модели*, если область $\Theta \subset R^d$ изменения параметра имеет ненулевой объем. Примером модели может служить множество M_k источников без памяти в k-буквенном алфавите A. В этом случае областью изменения параметра является $(k-1)$-мерный симплекс со стороной 1 (параметр — набор вероятностей букв) и

$$P_\theta(x^n) = \theta_1^{r_1} \dots \theta_{k-1}^{r_{k-1}} (1 - \theta_1 - \dots - \theta_{r-1})^{r_k},$$

где r_i — число вхождений буквы a_i в слово x^n.

Если распределение $P_\theta(x)$ источника известно, то задачу минимизации избыточности решает кодирование с длинами $|f(x^n)|$ кодовых слов наиболее близкими к $-\log P_\theta(x^n)$. С другой стороны, для произвольного префиксного кодирования f, вследствии неравенства Крафта (4), найдется распределение $Q(x^n)$ такое, что $|f(x^n)| \geq -\log Q(x^n)$ для всех $x^n \in A^n$. Из этого следует, что задачу кодирования неизвестного источника удобно разделить на две части: во-первых, выбор распределения $Q(x^n)$ и, во-вторых, построение кодирования с длинами кодовых слов близкими к $-\log Q(x^n)$. Вторая задача была рассмотрена в предыдущих параграфах. Соответственно разделим избыточность кодирования на две части: избыточность модели

$$D_n(P_\theta||Q) = E_{P_\theta} \log \frac{P_\theta(X^n)}{Q(X^n)}, \tag{11}$$

зависящую от выбора целевого распределения Q, и избыточность метода кодирования

$$r_n(f, Q) = E_{P_\theta}(|f(X^n)| + \log Q(X^n)).$$

Тогда из (7) имеем

$$R_n = \frac{1}{n}(D_n(P_\theta||Q) + r_n(f, Q)).$$

Из выпуклости вверх функции $\log t$ следует, что избыточность модели неотрицательна, т. е.

$$D_n(P_\theta||Q) \geq 0. \tag{12}$$

Теорема Шеннона утверждает, что избыточность метода кодирования $\frac{r_n(f,Q)}{n}$ на букву исходного сообщения можно сделать сколь угодно малой. В частности, это достигается с помощью арифметического кодирования.

Задача универсального кодирования, впервые поставленная Б. П. Фитингофом [31], состоит в нахождении распределения $Q^0(x^n)$, минимизирующего максимальную по всем $\theta \in \Theta$ избыточность модели

$$\inf_Q \sup_{\theta\in\Theta} D_n(P_\theta||Q) = \sup_{\theta\in\Theta} D_n(P_\theta||Q^0). \tag{13}$$

Рассмотрим произвольную вероятностную меру $\omega(\theta)$ на множестве Θ. Поскольку $\omega(\theta) \geq 0$ и $\int_\Theta d\omega(\theta) = 1$, справедливо равенство

$$\sup_{\theta\in\Theta} D_n(P_\theta||Q) = \sup_\omega \int_\Theta D_n(P_\theta||Q)d\omega(\theta). \tag{14}$$

Известно (см. [11]), что

$$\inf_Q \sup_\omega \int_\Theta D_n(P_\theta||Q)d\omega(\theta) = \sup_\omega \inf_Q \int_\Theta D_n(P_\theta||Q)d\omega(\theta), \tag{15}$$

причем $\sup\inf$ и $\inf\sup$ достигаются одновременно на некоторых распределении Q^0 и мере ω_0. Из (14) следует, что распределение Q^0, решающее задачу (15), минимизирует избыточность модели в смысле задачи (13).

Н. Мерхев и М. Федер [80] показали, что распределение $Q^0(x^n)$ содержится в параметрическом семействе M, если область Θ изменения параметра — выпуклая.

Пусть $\omega(\theta)$ — произвольная вероятностная мера и распределение $Q^\omega(x^n)$ определено равенством

$$Q^\omega(x^n) = \int_\Theta P_\theta(x^n)d\omega(\theta).$$

Из определения (11) величины $D_n(P_\theta||Q)$ следуют равенства

$$D_n(P_\theta||Q) = D_n(P_\theta||Q^\omega) + E_{P_\theta} \log \frac{Q^\omega(X^n)}{Q(X^n)},$$

$$\int_\Theta E_{P_\theta} \log \frac{Q^\omega(X^n)}{Q(X^n)} d\omega(\theta) = D_n(Q^\omega||Q).$$

Тогда

$$\int_\Theta D_n(P_\theta||Q)d\omega(\theta) = \int_\Theta D_n(P_\theta||Q^\omega)d\omega(\theta) + D_n(Q^\omega||Q). \tag{16}$$

Величину $D_n(P_\theta||Q^\omega)$ можно рассматривать как среднюю (относительно меры ω) избыточность модели. Асимптотическое поведение $D_n(P_\theta||Q^\omega)$ при $n \to \infty$ было исследовано Б. Кларком и А. Барроном [49]. Поскольку интеграл $\int_\Theta D_n(P_\theta||Q^\omega)d\omega(\theta)$ не зависит от распределения Q, а величина $D_n(Q^\omega||Q)$ неотрицательна (12) и равна 0 при $Q = Q^\omega$, то из (16) получаем

$$\inf_Q \int_\Theta D_n(P_\theta||Q)d\omega(\theta) = \int_\Theta D_n(P_\theta||Q^\omega)d\omega(\theta) = I(\Theta, A^n), \tag{17}$$

где последнее равенство является определением взаимной информации $I(\Theta, A^n)$ между областью Θ изменения параметра с вероятностной мерой $\omega(\theta)$ и множеством A^n слов с условными вероятностями, заданными равенством $Pr(X^n = x^n|\theta) = P_\theta(x^n)$. Величина $\sup_\omega I(\Theta, A^n)$ называется *пропускной способностью* канала связи. Известно (см., например, [3]), что супремум величины $I(\Theta, A^n)$ достигается на такой мере

$\omega_0(\theta)$, что избыточность модели $D_n(P_\theta||Q^{\omega_0})$ не зависит от $\theta \in \Theta$. Тогда из (14), (15) и (17) получаем соотношения, которые известны как теорема о равенстве избыточности и пропускной способности:

$$\inf_Q \sup_{\theta\in\Theta} D_n(P_\theta||Q) = \sup_\omega \int_\Theta D_n(P_\theta||Q^\omega)d\omega(\theta) = D_n(P_\theta||Q^{\omega_0}).$$

Эта теорема впервые была опубликована Б. Я. Рябко [16] и затем Л. Дэвиссоном и А. Леон-Гарсия [52]. Таким образом, распределение $Q^{\omega_0} = Q^0$ минимизирует избыточность модели.

Введем обозначение $\alpha_n(M) = D_n(P_\theta||Q^0)$, где распределение Q^0 минимизирует избыточность модели M. Для множества M_k источников без памяти в k-буквенном алфавите получены следующее результаты. Сначала Б. П. Фитингоф [31] показал, что $\alpha_n(M_k)/n \to 0$ при $n \to \infty$. Затем Р. Е. Кричевский [8, 9, 10] и независимо Л. Дэвисон с соавторами [53] показали, что $\alpha_n(M_k) = \frac{k-1}{2}\log n(1+o(1))$.

Существенным шагом в исследовании задачи универсального кодирования явилась работа Р. Е. Кричевского и В. К. Трофимова [67], в которой предложена вероятностная мера $\omega'(\theta)$ на $(k-1)$-мерном симпликсе со стороной 1:

$$d\omega'(\theta) = \frac{\Gamma(k/2)}{\pi^{k/2}}\theta_1^{-1/2}\ldots\theta_{k-1}^{-1/2}(1-\theta_1-\ldots-\theta_{k-1})^{-1/2}d\theta,$$

где Γ — гамма-функция Эйлера. Распределение

$$Q^{\omega'}(x^n) = \frac{\Gamma(k/2)\prod_{i=1}^k \Gamma(r_i+1/2)}{\pi^{k/2}\Gamma(n+k/2)}, \tag{18}$$

где r_i — число вхождений буквы a_i в слово x^n, является приближением к оптимальному на M_k распределению Q^0 в том смысле, что

$$\lim_{n\to\infty} \frac{\sup_{\theta\in\Theta} D_n(P_\theta||Q^{\omega'})}{D_n(P_\theta||Q^0)} = 1. \tag{19}$$

Источники, имеющие распределение со свойством (19), называют *универсальными* (см. [109]). Более того, Б. Кларк и А. Баррон [49] доказали, что для любой внутренней точки $\theta \in \Theta$ при $n \to \infty$ верно соотношение

$$D_n(P_\theta||Q^{\omega'}) - D_n(P_\theta||Q^0) \to 0. \tag{20}$$

Впоследствии с помощью распределения $Q^{\omega'}$ К. Ксай и А. Баррон [119] установили, что

$$\alpha_n(M_k) = \frac{k-1}{2}\log\frac{n}{2\pi e} + \log\frac{\pi^{k/2}}{\Gamma(k/2)} + o(1). \tag{21}$$

Для множества M_k^r марковских источников порядка r В. К. Трофимов [28] показал, что

$$\alpha_n(M_k^r) = (1/2)r^k(k-1)(1+o(1))\log n. \tag{22}$$

Й. Риссанен обобщил этот результат на произвольную модель M размерности d, удовлетворяющую некоторым дополнительным условиям гладкости распределения $P_\theta(x^n)$ как функции от θ. В работе [89] он показал, что

$$\alpha_n(M) = \frac{d}{2}\log n + O(1). \tag{23}$$

Кроме того, в работе [90] Й. Риссанен доказал теорему о строгой оптимальности распределения Q^0. А именно, если d — размерность модели, то для произвольного распределения Q и почти всех значений параметра θ выполнено неравенство

$$\limsup_{n\to\infty} \frac{D_n(P_\theta||Q)}{\log n} \geq d/2.$$

Обобщение последнего результата содержится в работе Н. Мерхева и М. Федера [80].

8. Распределение наибольшего правдоподобия

Задачу нахождения распределения $Q(x)$ минимизирующего избыточность кодирования неизвестного источника из параметрического семейства M можно ставить и по другому. Естественно предполагать, что сообщение x^n было порождено источником с параметром $\hat{\theta}(x^n) \in \Theta$ таким, что $P_{\hat{\theta}(x^n)}(x^n) = \max_{\theta\in\Theta} P_\theta(x^n)$. Рассмотрим множество M_k источников без памяти, т. е. область Θ изменения параметра является $(k-1)$-мерным симплексом со стороной 1. Величина $F(x^n) = -\log P_{\hat{\theta}(x^n)}(x^n)$ называется *эмпирической энтропией* слова x^n. Очевидно, что

$$F(x^n) = \sum_{i=1}^{k} \frac{r_i}{n} \log \frac{n}{r_i},$$

где r_i — число вхождений буквы a_i в слово x^n. Известно (см. [49]), что

$$E_{P_\theta} \log \frac{P_{\hat{\theta}(X^n)}(X^n)}{P_\theta(X^n)} = \frac{k-1}{2} \log e + o(1). \tag{24}$$

Последнее равенство не противоречит (12), поскольку $\sum_{x^n\in A^n} P_{\hat{\theta}(x^n)}(x^n) > 1$.

Для кодирования неизвестного сообщения Ю. М. Штарьков [37] предложил использовать распределение $Q^*(x^n)$ наибольшего правдоподобия (максимальной вероятности), определенное равенством

$$\max_{x^n\in A^n} \log \frac{P_{\hat{\theta}(x^n)}(x^n)}{Q^*(x^n)} = \min_{Q} \max_{x^n\in A^n} \log \frac{P_{\hat{\theta}(x^n)}(x^n)}{Q(x^n)}.$$

Пусть $K_n(M) = \sum_{x^n\in A^n} P_{\hat{\theta}(x^n)}(x^n)$. Покажем, что

$$Q^*(x^n) = \frac{P_{\hat{\theta}(x^n)}(x^n)}{K_n(M)}. \tag{25}$$

Действительно

$$\log \frac{P_{\hat{\theta}(x^n)}(x^n)}{Q(x^n)} = \log \frac{P_{\hat{\theta}(x^n)}(x^n)}{Q^*(x^n)} + \log \frac{Q^*(x^n)}{Q(x^n)}.$$

Поскольку $Q^*(x^n)$ и $Q(x^n)$ — распределения, то найдется такое слово $x^n \in A^n$, что $Q^*(x^n) \geq Q(x^n)$. Тогда $\max_{x^n\in A^n} \log \frac{Q^*(x^n)}{Q(x^n)} \geq 0$ и равенство нулю достигается только при $Q^*(x^n) = Q(x^n)$, т. е. равенство (25) доказано. Величина $-\log Q^*(x^n)$ названа Й. Риссаненом [90] *стохастической сложностью* слова x^n в соответствующей модели M.

Пусть $\beta_n(M) = \log K_n(M)$. По определению параметра $\hat{\theta}(x^n)$ имеем неравенство $P_\theta(x^n) \le P_{\hat{\theta}(x^n)}(x^n)$. Тогда

$$D_n(P_\theta||Q^*) \le E_{P_\theta} \log \frac{P_{\hat{\theta}(X^n)}(X^n)}{Q^*(X^n)} = \beta_n(M), \tag{26}$$

т. е. $\alpha_n(M) \le \beta_n(M)$. Ю. М. Штарьков [37] получил асимптотику для величины $\beta_n(M)$ для множеств марковских источников конечного порядка и источников без памяти. В частности, им показано, что $\beta_n(M_k) = \frac{k-1}{2} \log n(1 + o(1))$ при $n \to \infty$. Й. Риссанен [91] получил асимптотику величины $\beta_n(M)$ для гладких параметрических семейств M размерности d:

$$\beta_n(M) = \frac{d}{2} \log n + c(M) + o(1), \tag{27}$$

где $c(M)$ — точно определенная константа. В частности,

$$\beta_n(M_k) = \frac{k-1}{2} \log \frac{n}{2\pi} + \log \frac{\pi^{k/2}}{\Gamma(k/2)} + o(1). \tag{28}$$

Из соотношений (23), (26) и (27) следует, что

$$\lim_{n\to\infty} \frac{\sup_{\theta\in\Theta} D_n(P_\theta||Q^*)}{D_n(P_\theta||Q^0)} = 1.$$

Таким образом, распределение Q^* определяет универсальный источник. Покажем, что для источников без памяти распределения Q^* и $Q^{\omega'}$ асимптотически совпадают в среднем. Справедливы равенства

$$E_{P_\theta}(\log Q^*(X^n) - \log Q^{\omega'}(X^n))$$

$$= E_{P_\theta} \log \left(\frac{P_\theta(X^n)}{Q^{\omega'}(X^n)} \frac{P_{\hat{\theta}(X^n)}(X^n)}{P_\theta(X^n)} \frac{Q^*(X^n)}{P_{\hat{\theta}(X^n)}(X^n)} \right)$$

$$= D_n(P_\theta||Q^{\omega'}) + E_{P_\theta} \log \frac{P_{\hat{\theta}(X^n)}(X^n)}{P_\theta(X^n)} - \beta_n(M_k).$$

Отсюда и из соотношений (20), (21), (24), (28) для произвольной внутренней точки θ симплекса Θ получаем

$$E_{P_\theta}(\log Q^*(X^n) - \log Q^{\omega'}(X^n)) = \alpha_n(M_k) - \beta_n(M_k) + \frac{k-1}{2} \log e + o(1) = o(1).$$

9. Кодирование на основе статистики сообщений

На практике статистические характеристики источников, порождающих сообщения, чаще всего неизвестны. Сведения о свойствах источника как правило извлекают из самого сообщения и затем их явно или неявно используют в процессе кодирования. Кодирование очередной буквы сообщения происходит на основе статистики некоторой части сообщения, которая называется *окном*. Окно, соответствующее i-ой букве сообщения, будем обозначать через w_i. В качестве окна можно использовать все сообщение: $w_i = x$. Такое кодирование называется двухпроходным: на первом

проходе определяется статистика сообщения, на втором — сообщение кодируется на основе этой статистики. В этом случае задержка передачи сообщения равняется длине всего сообщения и помимо закодированного сообщения необходимо дополнительно передавать кодер (функцию, осуществляющую кодирование) или статистику, на основе которой он был построен. Необходимость в дополнительной информации отсутствует, если использовать универсальное кодирование. Однако непосредственное применение оптимального универсального кодирования для сжатия достаточно больших сообщений невозможно, поскольку трудоемкость построения префиксного кодирования с длинами кодовых слов, равных $-\log Q^0(x^n)$, экспоненциально растет как функция от длины сообщения. Метод универсального кодирования с полиномиальной трудоемкостью был предложен В. Ф. Бабкиным и Ю. М. Штарьковым [1, 101]. Избыточность этого кодирования вдвое больше избыточности оптимального кодирования. В частности, $R_n = \frac{(k-1)\log n}{n}(1 + o(1))$ для источников без памяти. Аналогичные методы кодирования были предложены также Т. Линчем [74], Л. Дэвиссоном [51] и Т. Ковером [50]. Эти универсальные методы кодирования можно также рассматривать как разновидность блочного кодирования: достаточно разделить сообщение на части одинаковой длины и кодировать их независимо. Из работ В. К. Трофимова [27] и Дж. Лоуренса [71] известно универсальное равномерное по выходу кодирование с полиномиальной трудоемкостью, избыточность которого в конечное число раз больше избыточности оптимального универсального кодирования.

Использование удлиняющегося окна $w_i = x^{i-1}$ также позволяет обойтись без передачи дополнительной информации. В этом случае при кодировании очередной буквы используются все предыдущие. В частности, при кодировании источников без памяти $(i+1)$-ую букву сообщения кодируют, основываясь на распределении вероятностей

$$P(a_j) = \frac{r_j(w_i) + \varepsilon}{|w_i| + k\varepsilon}, \tag{29}$$

где $r_j(w_i)$ — число вхождений буквы a_i в окне w_i длины $|w_i|$. Величина ε, $\varepsilon \geq 0$, называется сдвигом. Положительный сдвиг позволяет избежать неопределенности при кодировании первых вхождений букв в сообщение. Из формулы (18) следует, что

$$Q^{\omega'}(a_j|w_i) = \frac{Q^{\omega'}(x^{i-1}a_j)}{Q^{\omega'}(x^{i-1})} = \frac{r_j(w_i) + 1/2}{(i-1) + k/2},$$

т. е. асимптотически оптимальным является сдвиг $\varepsilon = 1/2$.

Д. Кнут [66] предложил конструкцию динамического кода Хаффмена. Каждая очередная буква x_i кодируется кодом Хаффмена, построенным по статистике окна $w_i = x^{i-1}$, а затем строится новый код Хаффмена, соответствующий окну w_{i+1}. Динамические коды Хаффмена были впоследствии исследованы Р. Галлагером [58] и Дж. Виттером [107]. Последний получил оценку $R \leq 2$ избыточности кодирования источников без памяти для специального класса этих кодов. Трудоемкость динамических кодов Хаффмена невелика — не более конечного числа операций над $\log n$-значными числами на букву сообщения. Ниже будет рассмотрена схема кодирования Лемпела–Зива, стандартное описание которой также использует удлиняющееся окно.

Практические алгоритмы сжатия данных обычно используют скользящее окно $w_i = x_{i-1-m}^{i-1}$ постоянной длины m, которое сдвигается на одну букву вправо при кодировании очередной буквы (такое кодирование часто называют адаптивным).

Широкое применение скользящего окна обусловлено тем, что, во-первых, ресурсы оперативной памяти кодирующих устройств ограничены и, во-вторых, реальные источники почти никогда не являются стационарными. Скользящее окно требует фиксированого объема памяти и непрерывно позволяет следить за изменяющимися статистическими свойствами сообщения. Р. Е. Кричевский [68] показал, что при использовании скользящего окна асимтотически оптимальным будет сдвиг $\varepsilon = 0,50922\ldots$ (см. формулу (29)).

Особенно эффективным при кодировании источников без памяти является использование скользящего окна и арифметического кодирования (вероятность очередной буквы определяется по формуле (29)). Б. Я. Рябко и А. Н. Фионов [24] показали, что надлежащий выбор точности арифметических операций и длины окна позволяет достигнуть избыточности $R \leq (1+\frac{1}{\ln 2})\frac{k}{m}$ при $V = O(m)$ и $T = O(\log m \log\log m)$, где m — длина скользящего окна.

Чтобы сократить число операций можно сдвигать окно не после поступления каждой очередной буквы, а через определенное число букв. Можно и совсем зафиксировать окно. Фиксированное окно позволяет экономить память, поскольку вместо окна можно хранить только частоты букв в окне. Конструкция, объединяющая достоинства фиксированного и скользящего окна, — "мнимое" скользящее окно — предложена Б. Я. Рябко [22]. Отличие "мнимого" от реального скользящего окна состоит в том, что на каждом шаге из окна удаляется не последняя, а случайная буква. Это дает возможность хранить не само окно, а только частоты букв. Случайная буква может выбираться как с помощью источника случайных величин, так и на основе уже закодированной части сообщения, поскольку код сообщения хорошо аппроксимирует источник Бернулли с вероятностями $P(0) = P(1) = 1/2$. В [22] для сообщений, порожденных источником без памяти, доказано, что

$$\lim_{i\to\infty} Pr(v_1(w_i) = n_1, \ldots, v_k(w_i) = n_k) = Pr(r_1(w_i) = n_1, \ldots, r_k(w_i) = n_k),$$

где $r_j(w_i)$ — число вхождений буквы a_j в реальное скользящее окно $w_i = x_{i-1-m}^{i-1}$, а $v_j(w_i)$ — число вхождений буквы a_j в "мнимое" окно.

10. Выбор модели источника и принцип кратчайшего описания

В предыдущих разделах мы исследовали кодирование сообщений, порожденных известным источником или источником, принадлежащим некоторому параметрическому семейству. Теперь мы рассмотрим более сложную, но более реальную задачу выбора параметрического семейства, в рамках которого следует рассматривать сообщение, чтобы получить наиболее короткий код сообщения. Предположим, что имеется счетное семейство источников $M_r = \{P_\theta(x) : \theta \in \Theta \subset R^{d_r}\}$. Будем считать, что множества M_r вложены друг в друга $M_1 \subset M_2 \subset \ldots \subset M_r \subset \ldots$ и соответственно $d_1 < d_2 < \ldots < d_r < \ldots$. Примером такой последовательности могут служить, например, множества марковских источников конечного порядка. Задача определения наименьшего семейства M_r, содержащего нужный источник, имеет смысл и если множество источников конечно, поскольку избыточность оптимального универсального кодирования линейно зависит от размерности параметра d_r модели M_r (см. формулу (23)).

Пусть имеется некоторый префиксный код натурального ряда, закодируем этим кодом индексы моделей. Пусть $l(r)$ — длина кода индекса модели M_r. Код сообщения можно составить из двух частей: кода индекса модели M_r и универсального

на модели M_r кода сообщения. Предложенный Й. Риссаненом [88] принцип кратчайшего описания состоит в выборе индекса r, минимизирующего длину $L(x)$ кода сообщения x, т. е.

$$L(x) = \min_r(l(r) - \log Q_r^0(x)), \tag{30}$$

где Q_r^0 — оптимальное универсальное распределение на M_r. Посуществу принцип кратчайшего описания был применен Б. Я. Рябко [19] при построении дважды универсального кодирования, которое является асимптотически оптимальным для всех классов марковских источников конечного порядка. Пусть r — порядок марковского источника, существует (см. (9)) префиксный код натурального ряда с длинами кодовых слов

$$l(r) = \log r + O(\log\log r). \tag{31}$$

Выбирая r по принципу кратчайшего описания, получим дважды универсальное кодирование с длинами кодовых слов как в (30), меньшими чем $l(i) - \log Q_i^0(x)$, где i — истинный порядок марковского источника. Очевидно, что для избыточности (модели) дважды универсального кодирования справедливы неравенства

$$\alpha_n(M_i) \le D_n \le \alpha_n(M_i) + l(i). \tag{32}$$

Из формул (23) и (31) видно, что избыточность дважды универсального кодирования асимптотически эквивалентна избыточности оптимального кодирования источников из M_i. Аналогичный результат получается, если в качестве кода для слова x выбирать слово длины $-\log Q(x)$, где

$$Q(x) = \sum_{k=1}^{\infty} 2^{-l(r)} Q_r^0(x). \tag{33}$$

Действительно

$$-\log Q(x) \le -\log(\max_r(2^{-l(r)}Q_r^0(x))) = \min_r(l(r) - \log Q_r^0(x)).$$

Л. Дэвиссон [51] предложил называть кодирование f *сильно универсальным* для источников из M, если $\sup_{X\in M} R_n(f,X) \to 0$ при $n\to\infty$, и *слабо универсальным*, если для произвольного источника $X \in M$ верно, что $R_n(f,X)\to 0$ при $n\to\infty$. Из формул (23) и (32) следует, что дважды универсальное кодирование является только слабо универсальным для всех марковских источников конечного порядка, в то время как оптимальное кодирование для марковских источников фиксированного порядка сильно универсально (см. раздел 7). Из (23) следует, что сильно универсальное кодирование для всех марковских источников конечного порядка невозможно, так как размерность модели неограничена.

Наиболее простое кодирование, использующее принцип кратчайшего описания предложили Р. Ньюход и Р. Шилдс [84]. Разделим сообщения x^n на блоки длины m. Составим словарь U_m из всех встречающихся в сообщении блоков. Каждый блок в сообщении x^n будем кодировать его номером в словаре. Естественно код сообщения помимо кодов блоков должен содержать словарь. Длина кода $L_m(x^n)$ сообщения x^n удовлетворяет равенству

$$L_m(x^n) = |U_m|m\lceil\log k\rceil + \frac{n}{m}\log|U_m|,$$

поскольку для записи слова длины m в k-буквенном алфавите достаточно $m\lceil \log k \rceil$ битов. Выберем длину блока $m(n)$, минимизирующую величину $L_m(x^n)$. Как показано в [84] это кодирование с длиной блока $m(n)$ является слабо универсальным для марковских источников.

Размерность параметра d_r семейства марковских источников порядка r растет экспоненциально в зависимости от r, а именно $d_r = k^r(k-1)$. Следовательно избыточность оптимального кодирования быстро растет в зависимости от порядка источника. Экспоненциально от порядка источника растет и объем памяти, требуемый для реализации метода (если кодовые слова строить с помощью явно использующих статистику методов кодирования).

Выбор в качестве множества моделей семейства источников-деревьев (см. раздел 1) позволяет находить более адекватную, а значит и более эффективную модель для кодирования каждого сообщения. Марковский источник r-го порядка может иметь много эквивалентных состояний, сокращение которых позволяет значительно уменьшить размерность модели. Ф. Виллемс, Ю. М. Штарьков и Ч. Чокенс в работе [112] предложили алгоритм построения взвешенного контекстного дерева, который в классе источников-деревьев позволяет найти близкое к оптимальному распределение $Q(x)$ такое, что

$$D_n(P_\theta||Q) = \frac{|\theta|}{2}\log n + O(1) \tag{34}$$

при $n \to \infty$, где $|\theta|$ — размерность модели источника-дерева, которой принадлежит источник с распределением P_θ, равная произведению числа листьев минимального дерева на размер алфавита без единицы.

Предлагаемый в [112] метод состоит в выборе некоторой вероятностной меры на множестве деревьев ограниченной высоты и определения затем целевого распределения согласно равенству (33). Рассмотрим множество U всех полных двоичных (будем рассматривать двухбуквенный алфавит) деревьев глубины не более $d > 0$. Пусть T' — множество листьев дерева T, а $x(s)$ — слово, состоящее из букв двоичного слова x, непосредственно следующих за вхождениями подслова s, $|s| \le d$ в слове x. Определим

$$P_T(x^n) = \prod_{s \in T'} Q^{\omega'}(x(s)), \tag{35}$$

где распределение $Q^{\omega'}$ задано равенством (18) и $T \in U$. Рассмотрим произвольное распределение $P_W(T)$ вероятностей на множестве U, т. е. $P_W(T) \ge 0$ и $\sum_{T \in D} P_W(T) = 1$. Пусть

$$Q_W(x^n) = \sum_{T \in U} P_W(T) P_T(x^n). \tag{36}$$

Тогда для произвольного источника-дерева с деревом T и распределением $P_\theta(x)$ справедливы неравенства

$$\log \frac{P_\theta(x^n)}{Q_W(x^n)} \le \log \frac{P_\theta(x^n)}{P_W(T)P_T(x^n)} \le \log \frac{1}{P_W(T)}$$

$$+ \sum_{s \in T'} \log \frac{P_\theta(x^n(s))}{Q^{\omega'}(x^n(s))} \le \log \frac{1}{P_W(T)} + |T'| \sum_{s \in T'} \frac{\log |x^n(s)| + 2}{2|T'|}$$

$$\le \log \frac{1}{P_W(T)} + \frac{|T'|}{2} \log \frac{n}{|T'|} + c,$$

где $c > 0$ — некоторая константа, $|\theta| = |T'|$ — размерность модели. Здесь первое неравенство следует из (36), второе — из (35), третье — из неравенства (P_θ — распределение произвольного источника Бернулли)

$$\log \frac{P_\theta(x^n)}{Q^{\omega'}(x^n)} \le \frac{1}{2}\log n + 1,$$

доказанного в [37] (см. также формулу (21)), четвертое неравенство следует из выпуклости вверх функции $\log t$. Поскольку U — конечное множество, то даже использование равномерного распределения $P_W(T) = 1/|U|$ позволяет получить распределение $P_W(x^n)$, избыточность которого отличается от избыточности оптимального в классе источников-деревьев распределения не более чем на некоторую константу. Ф.Виллемс с соавторами в работе [112] предложили распределение $P_W(U)$, минимизирующее эту константу асимптотически, и эффективное даже при малых n: $\log \frac{P_\theta(x^n)}{P_W(x^n)} \le n + 2|\theta| - 1$.

Кроме того, они предложили рекуррентную процедуру для получения этого распределения:
$P_W(x(s)) = Q^{\omega'}(x^n(s))$, если $|s| = d$;
$P_W(x(s)) = \frac{1}{2}(Q^{\omega'}(x^n(s)) + P_W(x(0s))P_W(x(1s)))$ в остальных случаях.

Распределение $P_W(x) = P_W(x(\emptyset))$ — искомое.

Ф. Виллемс с соавторами в работе [113] показали, что аналогичный метод эффективен и для других, более богатых по сравнению с источниками-деревьями множеств моделей источников и даже, как показал Ф. Виллемс [111], для множества неограниченных по высоте источников-деревьев.

Предложенный Й. Риссаненом [89] алгоритм "Контекст" дает возможность (см. [90, 109]) найти распределение $Q(x)$, близкое к оптимальному в классе источников-деревьев в смысле формулы (34) путем непосредственного построения источника-дерева по сообщению. Опишем быстрый вариант [92] алгоритма "Контекст" для двухбуквенного алфавита. Сначала построим рекуррентно некоторое множество $B(x^n)$ подслов слова x^n и множество контекстов $C(x^n)$, состоящее из начал этих подслов без последней буквы. Обозначим через $\delta(0|s)$ и $\delta(1|s)$ — индексы контекста s, принимающие значения 0 и 1.

Положим $B(\emptyset) = C(\emptyset) = \{\emptyset\}$ и $\delta(0|\emptyset) = \delta(1|\emptyset) = 1$. Пусть множество $B(x^{n-1})$ построено. Найдем наибольшее целое число $i \ge 0$ такое, что подслово x_{n-i}^n содержится в $B(x^{n-1})$. Тогда $B(x^n) = B(x^{n-1}) \cup \{x_{n-i-1}^n\}$. Если контекст $s = x_{n-i-1} \dots x_{n-1}$ не содержится в $C(x^{n-1})$, то определим $\delta(x_n|s) = 0$ и $\delta(\lambda|s) = 1$, где $\lambda \ne x_n$. Если $s \in C(x^{n-1})$, то изменим индекс: $\delta(x_n|s) = 0$. Затем определим $C(x^n) = C(x^{n-1}) \cup \{s\}$ и перейдем к следующей букве x_{n+1}. Можно показать, что множество $B(x^n)$ содержит все суффиксы и префиксы своих элементов. Например, пусть

$$x = 000011001100. \tag{37}$$

Тогда $B(x) = \{\emptyset, 0, 00, 000, 0000, 1, 11, 10, 100, 01, 011, 110, 1100\}$. Соответствующее множество контекстов $C(x) = \{\emptyset, 0, 00, 000, 10, 01, 11, 110\}$ можно представить в виде дерева. Оно изображено на рис. 2, контекст s задает путь к соответствующей ему вершине 0 — налево, 1 — направо, причем контекст читается справа налево. Справа от каждой вершины, соответствующей контексту s, указан итоговый индекс $\delta(0|s)$ и слева — $\delta(1|s)$.

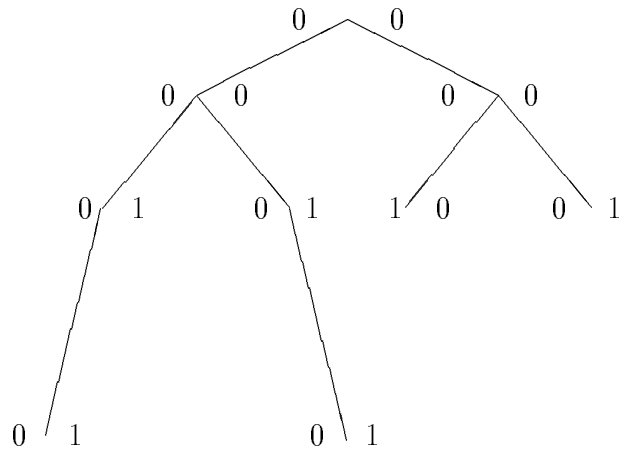

Рис. 2.

К каждому листу дерева T_1, соответствующему множеству контекстов $C(x)$, добавим по два листа и затем добавим еще недостающие листья так, чтобы дерево стало полным. Каждому листу τ полученного дерева T_2 припишем вес $c(0|\tau) = c(1|\tau) = 1$. Каждой внутренней вершине $s \in T_2$ припишем вес

$$c(\lambda|s) = c(\lambda|0s) + c(\lambda|1s) - \delta(\lambda|s),$$

где $\lambda = 0, 1$. Вес $c(\lambda|s)$ отличается от количества символов λ, встречающихся с контекстом s в слове x не более чем на 1. Дерево T_2, соответствующее слову (37), изображено на рис. 3, слева от вершины s указан вес $c(0|s)$, справа — $c(1|s)$.

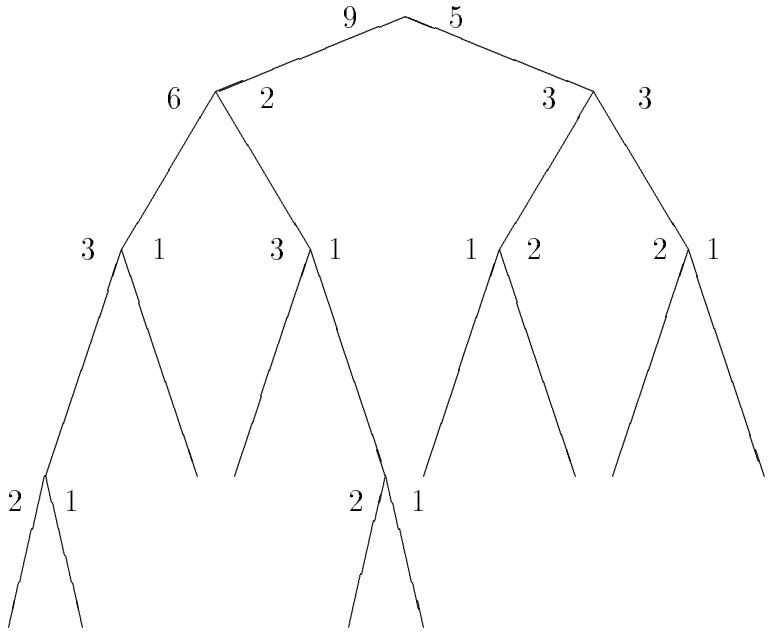

Рис. 3.

Сократим некоторые узлы дерева T_2 так, чтобы слово x в полученной модели источника-дерева имело наименьшую стохастическую сложность. Пусть $c(s) = c(0|s) + c(1|s)$. Тогда в соответствии с формулой (28) стохастическая сложность $L(s)$

(длина кода, удовлетворяющего критерию наибольшего правдоподобия) слова $x(s)$ равняется

$$L(s) = c(0|s)\log\frac{c(s)}{c(0|s)} + c(1|s)\log\frac{c(s)}{c(1|s)} + \frac{1}{2}\log\frac{c(s)\pi}{2}.$$

Оптимальный набор состояний-контекстов будем выбирать рекуррентно, начиная с листьев дерева T_2. Определим $J(s) = L(s)$ для всех листов, а для внутренних вершин дерева положим

$$J(s) = \min\{L(s), J(0s) + J(1s)\}.$$

Если первый элемент меньше или равен второму, то из дерева T_2 удалим потомков вершины s. Полученное дерево T — искомое. Распределение $Q(x^n) = \prod_{i=1}^{n} Q(x_i|x^{i-1})$, индуцированное деревом T, определяется в соответствии с формулой (29):

$$Q(x_i|x^{i-1}) = Q(x_i|s) = \frac{c(x_i|s) + 1/2}{c(s) + 1},$$

где s — суффикс x^{i-1}, являющийся листом дерева T. Распределение $O(x^n)$ является близким к оптимальному в классе источников-деревьев в смысле формулы (33).

Как показано Й. Риссаненом [92] рациональная организация выполнения алгоритма "Контекст" требует $O(n\log\log n)$ арифметических операций с числами длины $O(\log n)$ при кодировании слова длины n.

Арифметическое кодирование позволяет строить эффективные коды слов, которые реализуют распределения, построенные как алгоритмом "Контекст", так и с помощью взвешенного контекстного дерева.

11. Преобразование Барроуза–Уилера

Другим практически полезным методом кодирования последовательности, основанном на использовании контекстов, является преобразование Барроуза–Уилера [45]. Пусть x^n — некоторое слово в двоичном алфавите, превратим его в цикл, определив $x_0 = x_n, x_{-1} = x_{n-1}, \ldots, x_{-i} = x_{n-i}, \ldots$. Каждой букве x_i поставим в соответствие контекст $s(x_i)$ длины n: $s(x_i) = x_{i-1}x_{i-2}\ldots x_{i-n}$. Упорядочим контексты лексикографически (читая контекст справа — налево). *Преобразованием Барроуза-Уилера* $BW(x^n)$ называется слово длины n, составленное из букв слова x^n в порядке их контекстов, т. е. $BW(x^n) = x_{i_1}x_{i_2}\ldots x_{i_n}$, где $s(x_{i_1}) \le s(x_{i_2}) \le \ldots \le s(x_{i_n})$. Другими словами, рассмотрим матрицу вращений слова x^n и расставим строки матрицы в лексикографическом (начиная с конца) порядке. Первый столбец получившейся матрицы и есть слово $BW(x^n)$. Например, пусть $x = 110110100$. Тогда матрица вращений слова x имеет вид

$$\begin{array}{ccccccccc}
1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0.
\end{array}$$

После упорядочивания получаем матрицу

$$\begin{array}{ccccccccc}
1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1.
\end{array}$$

Тогда $BW(110110100) = 101111000$. Оказывается, что по слову $BW(x)$ можно восстановить всю упорядоченную матрицу. Во всех столбцах (как и в строках) матрицы содержится одинаковое число 0 и 1. Последний столбец вследствии лексикографического порядка строк имеет вид $00\ldots011\ldots1$, где число нулей и единиц известно. Поскольку строки зациклены, то пара, состоящая из последней и первой буквы каждой строки упорядоченной матрицы, содержится в слове x. Упорядочив множество пар мы получим предпоследний столбец упорядоченной матрицы. Аналогичным образом получая тройки, четверки и т. д. мы восстановим всю матрицу. Теперь чтобы найти исходное слово x нам нужен лишь номер строки упорядоченной матрицы совпадающий с x.

Если рассмотреть слово x, порожденное некоторым источником-деревом, то его преобразование $BW(x)$ будет состоять из последовательности $x(s_1)x(s_2)\ldots x(s_t)$, где $x(s_i)$ — подслово, порожденное в состоянии s_i, причем состояния s_i лексикографически упорядоченны.

Методы кодирования преобразования Барроуза–Уилера $BW(x)$ двоичного слова x, а также эффективные алгоритмы выполнения преобразования рассмотрены А. В. Кадачем [5]. Выполнение преобразования Барроуза–Уилера слова длины n требует $O(n\log^2 n)$ операций над битами, а обратное преобразование требует $O(\frac{n\log n}{H})$ операций над битами в среднем по сообщениям, порожденным стационарным источником с энтропией H. Эмпирические исследования [5] показывают, что слова $BW(x)$ можно эффективно кодировать интервальными методами, которые будут рассмотрены ниже.

12. Схемы кодирования Лемпела-Зива

Помимо описанных в предыдущих параграфах методов универсального кодирования, которые в явном виде используют контексты и статистику сообщений, существуют алгоритмические методы сжатия данных, использующие статистические характеристики источников неявно.

Пусть $A = \{a_1, \ldots, a_k\}$ — некоторый алфавит и $x \in A^n$. Схема кодирования, предложенная А. Лемпелом и Я. Зивом в 1977 году [122] (в дальнейшем именуемая LZ77), состоит в разделении кодируемого слова $x^n \in A^n$ на подслова σ_i, $i = 1\ldots m$, по следующему правилу. Пусть начало слова x^n уже разделено на подслова, т. е. представляет собой конкатенацию подслов $\sigma_1\sigma_2\ldots\sigma_i$ и $x^n = \sigma_1\ldots\sigma_i x_{l_i}^n$. Выберем следующее подслово $\sigma_{i+1} = x_{l_i}^{r_i}$ как наиболее длинное начало остатка $x_{l_i}^n$, которое уже встречалось в $x_1^{r_i-1}$, т. е.

$$\sigma_{i+1} = x_{l_i}^{r_i} = x_{n_i}^{n_i+r_i-l_i},$$

где $n_i < l_i$. Кодом каждого подслова σ_{i+1} будет пара чисел $(n_i, r_i - l_i)$. Например, слово $(a_1a_2)a_2a_1a_2a_1a_1a_2a_1a_2a_1a_2$ разделяется на подслова $a_2, a_1a_2, a_1, a_1a_2a_1, a_2a_1a_2$ и кодируется последовательностью пар чисел $(2,1),(1,2),(1,1),(4,3),(3,3)$. Первое число в каждой паре целесообразно записывать в двоичном виде с использованием $\lfloor \log l_i \rfloor + 1$ битов, второе можно кодировать произвольным префиксным кодом чисел натурального ряда.

В схеме кодирования LZ77 можно использовать фиксированнное или скользящее окно длины w, что чаще всего и делается на практике (см., например, [5]). Тогда подслово σ_{i+1} выбирается как наиболее длинное подходящее начало остатка сообщения, которое содержится в окне. В этом случае для записи первого числа в каждой паре достаточно $\lceil \log w \rceil$ битов. Второе число в паре можно записывать с помощью произвольного префиксного кода натурального ряда, например, кода Левенштейна [13]. Из формулы (9) следует, что длина кода $L(\sigma_i)$ подслова σ_i удовлетворяет неравенству

$$L(\sigma_i) \le \log w + \log |\sigma_i| + 2 \log \log |\sigma_i| + c, \tag{38}$$

где $c > 0$ — некоторая константа и $|\sigma_i| = r_i - l_i + 1$ — длина подслова σ_i.

Известна модификация схемы LZ77, предложенная П. Бендером и Дж. Вольфом [43]. Она заключается в том, что после нахождения длиннейшего подходящего подслова σ_{i+1} в окне w_i нужно найти подходящее подслово $\sigma'_{i+1} \subset w_i$ — наиболее длинное, не считая σ_{i+1}. Вместо числа $|\sigma_{i+1}|$ нужно кодировать число $|\sigma_{i+1}| - |\sigma'_{i+1}|$. При декодировании, зная начало подслова σ_{i+1}, нетрудно найти в окне w_i наиболее длинное подслово σ'_{i+1}, совпадающее с началом подслова σ_{i+1}. Тогда длина подслова σ_{i+1} легко восстанавливается из равенства

$$|\sigma_{i+1}| = |\sigma'_{i+1}| + (|\sigma_{i+1}| - |\sigma'_{i+1}|).$$

Схема кодирования, предложенная А. Лемпелем и Я. Зивом в 1978 году [123] (в дальнейшем именуемая LZ78) отличается от описанной выше тем, что на каждом шаге выбирается наиболее длинное начало остатка $x_{l_i}^n$, которое совпадает с некоторым уже выделенным подсловом σ_j, $j < i$, и к нему добавляется еще одна буква, т. е. $\sigma_{i+1} = \sigma_j a_{p_i}$. Кодом подслова σ_{i+1} будет пара чисел (j, p_i). Например, слово $a_2a_1a_2a_1a_1a_2a_1a_2a_1$ разделяется на подслова $a_2, a_1, a_2a_1, a_1a_2, a_1a_2a_1$ и кодируется последовательностью пар чисел $(0,2),(0,1),(1,1),(2,2),(4,1)$.

Схему LZ78 удобно рассматривать как кодирование с динамическим словарем U. Сначала словарь U состоит из всех букв алфавита источника. На каждом шаге алгоритма отделяем от остатка кодируемого слова наиболее длинное слово $y \in U$ и добавляем в словарь U все слова вида ya_i. Словарь U можно произвольным образом нумеровать и удалять из него слова, все продолжения которых уже имеются в словаре.

Наиболее известная модификация этой схемы кодирования предложена Т. А. Велчем [110]. Она отличается от схемы LZ78 тем, что на каждом шаге в словарь добавляется только одно слово ya_i, где a_i следующая за словом y буква в кодируемом слове x.

Существует еще несколько вариантов схем Лемпела–Зива (самые известные [102, 120]), которые оптимизируют алгоритм поиска и выделения подслов, способы записи натуральных чисел, кодирование первого вхождения буквы в слово и т. п. В частности, проблема оптимизации разделения слова на подслова рассмотрена

А. В. Кадачем [5], а задача эффективного кодирования длин выделенных подслов исследована Е. И. Ситняковской [26]. Подробный обзор вариантов схем Лемпела–Зива сделал Т. Белл с соавторами [42].

Непосредственно из алгоритмов LZ77 и LZ78 следует, что выполнение декодирования значительно менее трудоемко, чем выполнение кодирования. Известны версии схем кодирования LZ77 и LZ78 (см. [5]), трудоемкость декодирования которых линейна относительно длины n сообщения, а трудоемость кодирования есть величина $O(n \log n)$.

Уже в 1978 г. А. Лемпел и Я. Зив [123] доказали, что схемы кодирования LZ77 и LZ78 являются слабо универсальными на множестве всех марковских источников с конечными алфавитом и множеством состояний. В дальнейшем оценки избыточности неоднократно уточнялись. Е. Плотник с соавторами [85] показали, что при $n \to \infty$

$$R_n(f^{78}, X) = O\Big(\frac{\log\log n}{\log n}\Big).$$

Здесь и далее f^{78} — кодирование, построенное по схеме LZ78 с удлиняющимся окном (основной вариант), X — произвольный марковский источник. Затем оценка избыточности была улучшена С. Савари [95]:

$$R_n(f^{78}, X) \le O\Big(\frac{1}{\log n}\Big).$$

Г. Лоучард и В. Зпанковский [73] установили, что эта оценка неулучшаема для источников Бернулли и нашли в этом случае асимптотику для $R_n(f^{78}, X)$ с точностью до эквивалентности. Для кодирования f_w^{77}, использующего схему LZ77 со скользящим окном длины w, Х. Морита и К. Кобояши [83] показали что при $w \to \infty$

$$R(f_w^{77}, X) = O\Big(\frac{\log\log w}{\log w}\Big).$$

Затем для варианта схемы LZ77 с фиксированным окном А. Дж. Винер [118] для марковских источников конечного порядка установили, что при $w \to \infty$

$$R(f_w^{77}, X) = \frac{H(X)\log\log w}{\log w}(1 + o(1)).$$

Для схемы LZ77 с удлиняющимся окном С. Савари [96] получил следующую оценку избыточности:

$$R_n(f^{77}, X) \le \frac{2H(X)\log\log n}{\log n}(1 + o(1)).$$

Кодирование f_w^{91}, использующее модификацию П. Бендера и Дж. Вольфа с фиксированным окном, оказалась асимптотически эффективнее основного алгоритма. А. Д. Винер и А. Дж. Винер [116] показали, что

$$R_n(f_w^{91}, X) = O\Big(\frac{1}{\log w}\Big).$$

Избыточность модификации Т. А. Велча асимптотически совпадает с избыточностью основной схемы LZ78 (см. [95]).

Основные идеи, использованные при установлении оценок избыточности схем кодирования Лемпела–Зива, можно проследить на примере схемы LZ77 с фиксированным окном длины w. Пусть X — некоторый марковский источник. В данном случае удобно рассмотреть последовательность X_n бесконечную в обе стороны. Обозначим через $N_l(X)$ наименьшее натуральное число N такое, что $X_1^l = X_{-N+1}^{-N+l}$. Ясно, что случайная величина $N_l(X)$ в среднем должна быть близка к $\frac{1}{P(X_1^l)}$ при $l \to \infty$. Поскольку

$$H(X) = \lim_{l\to\infty} \frac{1}{l} E \log \frac{1}{P(X_1^l)},$$

можно предположить, что $\frac{\log N_l(X)}{l} \approx H(X)$. Действительно А.Д.Винер и Я.Зив [115] доказали, что по вероятности

$$H(X) = \lim_{l\to\infty} \frac{\log N_l(X)}{l}. \tag{39}$$

Рассмотрим подслово σ_1, которое является длиннейшим началом последовательности X_1^∞, содержащееся в X_{-w+1}^0. Из совпадения событий $\{N_l > w\} = \{|\sigma_1| < l\}$ и равенства (39) можно заключить (см. [115]), что по вероятности

$$H(X) = \lim_{w\to\infty} \frac{\log w}{|\sigma_1(X)|},$$

Следовательно,

$$H(X) = \lim_{w\to\infty} \frac{\log w}{E|\sigma_1(X)|}. \tag{40}$$

Из выпуклости вверх функции $\log t$ и неравенства (40) получаем неравенства

$$E \log |\sigma_1(X)| \le \log(E|\sigma_1(X)|) \le \log\log w + c, \tag{41}$$

где $c > 0$ — константа.

Стоимостью кодирования называется отношение длины слова к длине кода. Поэтому справедливо подробно обсуждаемое в [118] равенство

$$C(f_w^{77}, X) = \lim_{i\to\infty} \frac{EL(\sigma_i(X))}{E|\sigma_i(X)|} = \frac{EL(\sigma_1(X))}{E|\sigma_1(X)|}, \tag{42}$$

где $L(\sigma_1)$ — длина кода подслова σ_1, а второе равенство следует из стационарности источника X. Тогда из соотношений (38)–(41) при $w \to \infty$ получаем верхнюю оценку избыточности

$$R(f_w^{77}, X) = C(f_w^{77}, X) - H(X) \le \frac{\log w + E(\log|\sigma_i(X)| + 2\log\log|\sigma_i(X)| + c)}{E|\sigma_1(X)|} - H(X)$$

$$= \frac{H(X)\log\log w}{\log w}(1 + o(1)).$$

13. Интервальное кодирование

Еще одним алгоритмическим методом сжатия данных является интервальное кодирование, отличающееся быстродействием и простотой реализации. Интервальное кодирование состоит в следующем: в исходной последовательности каждая буква

заменяется на число, равное количеству букв до предыдущего вхождения той же буквы. Например слово

$$(a_1a_2a_3)a_3a_3a_3a_2a_2a_2a_1a_1a_1a_3 \tag{43}$$

будет преобразовано в последовательность чисел (...)0004008006. Известны две модификации этого метода, позволяющие уменьшить стоимость кодирования. Первая из них была предложена Б. Я. Рябко [18] и названа им методом стопки книг. Этот метод отличается от интервального кодирования тем, что вместо числа всех букв между двумя одинаковыми указывается число различных букв между ними. Так, например, слово из (43) будет преобразовано в (...)0001002002. Метод стопки книг можно рассматривать как упрощение схемы кодирования LZ78. В этом случае словарь U состоит только из букв (или слов одинаковой длины), которые при поступлении очередной буквы перенумеровываются. Впоследствии метод стопки книг был переоткрыт П. Элайесом [55] и Дж. Бентли с соавторами [44].

Другая модификация интервального кодирования была предложена З. Арнавутом и С. Магливерасом [39]. Она заключается в том, что каждая буква исходного слова заменяется числом букв с большими номерами, разделяющих текущее и предыдущее включение буквы. Например, слово из (43) будет преобразовано в три последовательности, соответствующие трём различным буквам: a_3 : (...)0000, a_2 : (...)400, a_1 : (...)800. Декодирование нужно начинать с первой буквы алфавита, оставляя для других букв соответствующее количество пустых мест.

Для кодирования последовательности чисел, которая получается из исходной последовательности после применения интервального кода, можно использовать произвольный префиксный код для натуральных чисел. В частности, можно применить код Левенштейна [13].

Для интервального кодирования источника без памяти Б. Я. Рябко [18] получил оценку избыточности $R \leq 2\log\log k + c$, где k — объем алфавита источника и $c > 0$ — некоторая константа. Действительно, из равенства (9) следует оценка длины кода i-ой буквы слова x_i:

$$L(x_i) \leq \log N_i + 2\log\log N_i + c, \tag{44}$$

где N_i — расстояние до предыдущего вхождения буквы x_i в слове x и $c > 0$ — некоторая константа. Предположим что частота буквы близка к ее вероятности. Тогда аналогично (39) имеем $E\log N_i(X) = H(X)$. Поскольку $H(X) \leq \log k$, из неравенства (44) получаем неравенство

$$EL(x_i) \leq H(X) + 2\log\log k + c,$$

где $c > 0$ — некоторая константа. Из последнего неравенства и формулы (42) следует искомая оценка избыточности.

Ясно, что время кодирования и декодирования интервальными методами линейно относительно длины сообщения и объем используемой памяти конечен, если ограничен объем алфавита источника.

14. Кодирование текстов на естественных языках

Значительной частью реально сжимаемых данных являются тексты на естественных языках (русском, английском и др.). Описанные выше схемы кодирования

не используют в явном виде особенности текстов на естественных языках, в частности, не используют разделение текста на слова, набор которых весьма ограничен. Кроме того, на практике часто необходимо обеспечить произвольный доступ к текстовым данным, в то время как рассмотренные выше методы обеспечивают только последовательный доступ к сжатым данным. При кодировании больших текстов часто оказывается целесообразным составить словарь и кодировать слова целиком, рассматривая их как буквы нового алфавита — словаря. Такие методы изучались Э. Шварцем [98], А. Мофо [81] и Т. Беллом с соавторами [41].

Исследуя естественные языки, Г. К. Ципф [121] обнаружил, что если занумеровать слова естественного языка в порядке убывания частоты встречаемости, то вероятность слова будет приблизительно обратно пропорциональна его номеру. Для объяснения этой закономерности было выдвинуто несколько гипотез. В частности, Б. Мандельброт [75] показал, что если пробел между словами рассматривать как случайный символ, то будет выполняться закон Ципфа. Он же получил это распределение, исходя из предположения, что эволюционный процесс выбора длин слов может быть описан как случайное блуждание [76]. Б. Я. Рябко [16] показал, что распределение Ципфа близко к оптимальному универсальному распределению на множестве источников без памяти с упорядоченными вероятностями букв.

Предполагая, что вероятности букв подчиняются распределению Ципфа, нетрудно оценить энтропию источника. Пусть X — источник без памяти с вероятностями букв $p(a_i) = \frac{1}{i\gamma(k)}$, где k — число букв в алфавите и

$$\gamma(k) = \sum_{i=1}^{k} \frac{1}{i} = \ln k + c + O(1/k) \tag{45}$$

и c — постоянная Эйлера. Из (45) и известного неравенства

$$\sum_{i=1}^{k} \frac{\log i}{i} \geq \frac{\ln^2 k}{2 \ln 2} - 2$$

получаем

$$H(X) = \sum_{i=1}^{k} \frac{\log i\gamma(k)}{i\gamma(k)} = \log \gamma(k) + \frac{1}{\gamma(k)} \sum_{i=1}^{k} \frac{\log i}{i} \geq \frac{\log k}{2}.$$

Таким образом, применение пословного префиксного кодирования для текстов на естественных языках может сжать текст не более, чем в два раза лучше по сравнению с равномерным кодированием каждого слова $\lceil \log k \rceil / 2$ битами. М. Гутман [63] нашел черезвычайно простой код Хаффмена для источника с распределением Ципфа и $k = 2^{2^t}$, где $t > 0$ — целое. Код i-ой буквы состоит из двух частей: вторая — $Bin(i)$ (двоичная запись числа i), первая — двоичная запись длины слова $Bin(i)$ с использованием ровно t битов.

При словарном кодировании необходимо хранить словарь, что требует значительного объема памяти. Редко встречающиеся слова записывать в словарь нецелесообразно. Вопрос, какую часть слов хранить, а какую кодировать побуквенно, исследован М. П. Шаровой [34]. Оказывается, что наиболее эффективным является включение в словарь только $O(k/(\ln k)^{\varepsilon})$ (при $k \to \infty$) наиболее вероятных слов, где $\varepsilon > 0$ — некоторая константа.

15. Кодирование источников с низкой энтропией

Другим практически важным классом источников являются источники с низкой энтропией. Возможность построения более простых, а значит и менее трудоемких по сравнению с общим случаем, кодов для источников с низкой энтропией была замечена еще К. Шенноном [100], который предложил первый специальный код для последовательности редких событий. Последовательности редких событий порождаются двухбуквенными источниками Бернулли с малой вероятностью единицы $P(1) = p$. Они состоят из длинных серий нулей, разделенных редкими событиями — единицами.

К. Шеннон [100] предложил кодировать длины серий числами в $(2^m - 1)$-ичной системе, выделив один блок из m единиц как код запятой, разделяющей числа. Если выбрать $m = \lceil \log \frac{1}{p} \rceil$, то можно получить оценку избыточности кодирования $R = O(p \log \frac{1}{p})$ при $p \to 0$.

Одним из наиболее известных (см. [35, 56]) способов сжатия таких источников является кодирование длин серий высоковероятных символов с помощью префиксного кода чисел натурального ряда, что является по-существу частным случаем интервального кодирования в модификации З. Арнавута и С. Магливераса [39].

Оценим избыточность кодирования длин серий посредством кода Левенштейна [13]. Из равенства (9) заключаем, что длина кода i-ой серии удовлетворяет асимптотическому равенству

$$L_i = \log l_i + \log \log l_i (1 + o(1)), \tag{46}$$

когда длина серии $l_i \to \infty$. Поскольку среднее расстояние между единицами равняется числу $1/p$, то из (46) имеем неравенство

$$R = pEL_i - H \le p \log \log \frac{1}{p} (1 + o(1))$$

при $p \to 0$. С другой стороны из неравенства (10) можно получить, что

$$R \ge p \log \log \frac{1}{p} (1 + o(1))$$

при $p \to 0$.

Другой эффективный способ кодирования длин серий был предложен С. Голомбом [61]. Согласно этому методу исходная последовательность разделяется на блоки A_i, каждый из которых состоит из i нулей $(0 \le i \le l - 1)$ и следующей за ними единицей. Блок A_l содержит l нулей, где $l = \lceil 1/2 + \log \frac{1}{p} \rceil$. Блоки кодируются специальным префиксным кодом. Избыточность кода Голомба равняется $O(p)$ при $p \to 0$. Р. Галлагер и Д. ван Вурхис [59] показали, что если p таково, что $p^l + p^{l+1} \le 1 < p^l + p^{l-1}$, то код Голомба является оптимальным для префиксного кодирования длин серий. В работе В. Ф. Бабкина и Б. М. Книжного [2] предложен адаптивный вариант кода Голомба, позволяющий кодировать сообщения за один проход.

Б. Я. Рябко и М. П. Шарова [25, 33, 35] разработали несколько методов кодирования для марковских источников с низкой энтропией. Рассмотрим идею их метода на примере источника Бернулли с двухбуквенным алфавитом. Кодирование состоит из двух этапов. На первом этапе сообщение разбивается на блоки длины $l = \lceil 1/\sqrt{p} \rceil$. Блок из нулей кодируется одним нулем, если блок содержит хотя бы одну единицу, то его кодом будет весь блок с добавленной перед ним единицей. После

первого этапа длина сообщения сильно сокращается, а полученный код можно рассматривать как сообщение, порожденное марковским источником с согласованными вероятностями. На втором этапе к полученному после первого этапа слову применяется арифметический код, использующий вычисленные аналитически вероятности символов. Предложенный метод имеет в $1/\sqrt{p}$ раз большее быстродействие по сравнению с арифметическим кодированием при той же избыточности и используемой памяти, что и у арифметического кода.

Литература

[1] Бабкин В. Ф. Метод универсального кодирования независимых сообщений неэкспоненциальной трудоемкости // Проблемы передачи информации. 1971. Т. 7, Вып. 4. С. 13–21.

[2] Бабкин В. Ф., Книжный Б. М. Об адаптивном коде Голомба для длин серий// Труды X симпозиума по проблеме избыточности в информационных системах. Тез. докл. Ленинград. 1989. Ч. 2. С. 23–26.

[3] Галлагер Р. Теория информации и надежная связь. М.: Советское радио. 1974.

[4] Гоппа В. Д. Коды и информация// Успехи мат. наук. 1984. Т. 39, № 1. С. 77–120.

[5] Кадач А. В. Эффективные алгоритмы неискажающего сжатия текстовой информации: Дисс... канд. физ.-мат. наук. Новосибирск: Ин-т систем информатики им. А. П. Ершова, 1997.

[6] Колмогоров А. Н. Три подхода к определению понятия "количество информации" // Проблемы передачи информации. 1965. Т 1, Вып. 1. С. 3–11.

[7] Кричевский Р. Е. Длина блока, необходимая для получения заданной избыточности// Докл. АН СССР. 1966. Т. 171, № 1. С. 37–40.

[8] Кричевский Р. Е. Связь между избыточностью кодирования и достоверностью сведений об источнике // Проблемы передачи информации. 1968. Т. 4, Вып. 3. С. 48–57.

[9] Кричевский Р. Е. Лекции по теории информации. Новосибирск: Изд-во НГУ, 1970.

[10] Кричевский Р. Е. Оптимальное кодирование источника на основе наблюдений // Проблемы передачи информации. 1975. Т. 11, Вып. 1. С. 37–42.

[11] Кричевский Р. Е. Сжатие и поиск информации. М.: Радио и связь. 1989.

[12] Левенштейн В. И. О некоторых свойствах кодовых систем// Докл. АН СССР. 1961. Т. 140, № 6. С. 1274–1277.

[13] Левенштейн В. И. Об избыточности и замедлении разделимого кодирования натуральных чисел// Проблемы кибернетики. М.: Наука, 1968. Вып. 20. С. 173–179.

[14] Марков А. А. Введение в теорию кодирования. М: Наука. 1982.

[15] Потапов В. Н. Оценки избыточности кодирования последовательностей алгоритмом Лемпела–Зива// Дискрет. анализ и исслед. операций. Сер. 1. 1999. Т. 6, № 2. С. 70–81.

[16] Рябко Б. Я. Кодирование источника с неизвестными, но упорядоченными вероятностями// Проблемы передачи информации. 1979. Т. 15, Вып. 2. С. 71–77.

[17] Рябко Б. Я. Универсальное кодирование компактов// Докл. АН СССР. 1980. Т. 252, № 6. С. 1325–1328.

[18] Рябко Б. Я. Сжатие данных с помощью стопки книг// Проблемы передачи информации. 1980. Т. 16, Вып. 2. С. 16–21.

[19] Рябко Б. Я. Дважды универсальное кодирование// Проблемы передачи информации. 1984. Т. 20, Вып. 3. С. 24–28.

[20] Рябко Б. Я. Алгоритмический подход к задаче прогнозирования// Проблемы передачи информации. 1993. Т. 29, Вып. 2. С. 96–103.

[21] Рябко Б. Я. Эффективный метод кодирования источников информации, использующий алгоритм быстрого умножения// Проблемы передачи информации. 1995. Т. 31, Вып. 1. С. 3–12.

[22] Рябко Б. Я. Сжатие данных с помощью "мнимого скользящего окна"// Проблемы передачи информации. 1996. Т. 32, Вып. 2. С. 22–30.

[23] Рябко Б. Я., Фионов А. Н. Быстрый метод рандомизации сообщений// Проблемы передачи информации. 1997. Т. 33, Вып. 3. С. 3–14.

[24] Рябко Б. Я., Фионов А. Н. Эффективный метод арифметического кодирования для источников с большими алфавитами// Проблемы передачи информации. Принята к публикации.

[25] Рябко Б. Я., Шарова М. П. Быстрое кодирование низкоэнтропийных источников// Проблемы передачи информации. 1999. Т. 35, Вып. 1. С. 49–60.

[26] Ситниковская Е. И. Построение эффективных побуквенных кодов для словарных методов сжатия данных// Проблемы передачи информации. 1998. Т. 34, Вып. 2. С. 47–56.

[27] Трофимов В. К. Универсальное равномерное по выходу кодирование бернуллиевских источников// Методы дискретного анализа в теории кодов и схем: Сб. науч. тр. Новосибирск: Ин-т математики СОАН СССР. 1976. Вып. 29. С. 87–100.

[28] Трофимов В. К. Избыточность универсального кодирования произвольных марковских источников//Проблемы передачи информации. 1982. Т. 18, Вып. 2. С. 3–11.

[29] Фионов А. Н. Эффективный метод рандомизации сообщений на основе арифметического кодирования//Дискрет. анализ и исслед. операций. Сер. 1. 1997. Т. 4, № 2. С. 51–74.

[30] Фионов А. Н. Методы эффективной рандомизации сообщений, базирующиеся на омофонном и арифметическом кодировании: Дисс... канд. тех. наук. Новосибирск: СибГУТИ, 1998.

[31] Фитингоф Б. М. Оптимальное кодирование при неизвестной и меняющейся статистике сообщений// Проблемы передачи информации. 1966. Т. 2, Вып. 2. С. 3–11.

[32] Ходак Г. Л. Оценки избыточности при пословном кодировании сообщений, порожденных бернуллиевскими источниками// Проблемы передачи информации. 1972. Т. 8, Вып. 2. С. 21–32.

[33] Шарова М. П. Быстрое кодирование марковских источников с малой энтропией// Дискрет. анализ и исслед. операций. Сер. 1. 1998. Т. 5, № 4. С. 81–96.

[34] Шарова М. П. Влияние объема словаря на степень сжатия текста// Дискрет. анализ и исслед. операций. Сер. 1. 1999. Т. 6, № 1. С. 86–96.

[35] Шарова М. П. Эффективные методы кодирования низкоэнтропийных источников: Дисс... канд. физ.-мат. наук. Новосибирск: Ин-т математики им. С. Л. Соболева СО РАН, 1999.

[36] Штарьков. Ю. М. Обобщенные коды Шеннона// Проблемы передачи информации. 1984. Т. 20, Вып. 3. С. 3–16.

[37] Штарьков. Ю. М. Универсальное кодирование отдельных сообщений// Проблемы передачи информации. 1987. Т. 23, Вып. 3. С. 3–17.

[38] Ahlwede R., Han H. S., Kobayashi K. Universal coding of integers and unbounded search trees// IEEE Trans. Inform. Theory. 1997. V. 43, N 3. P. 669–683.

[39] Arnavut Z., Magliveras S. S. Block sorting and data compression// Proc. IEEE Data Compression Conference. Snowbird, Utah. 1997. P. 181–190.

[40] Barron A., Rissanen J., Yu B. The minimum description length principle in coding and modeling// IEEE Trans. Inform. Theory. 1998. V. 44, N 6. P. 2743–2760.

[41] Bell T. C., Cleary J. G., Witten I. H. Text compression. Prentice Hall. N.Y.:Englewood Cliffs. 1990

[42] Bell T. C., Witten I. H., Cleary J. H. Modeling for text compression // ACM Computing Surveys. 1989. V. 21, N 4. P. 557–591.

[43] Bender P. E., Wolf J. New asymptotic bound and improvements on the Lempel-Ziv data compression algorithm// IEEE Trans. Inform. Theory. 1991. V. 37, N 3. P. 721–729.

[44] Bentley J. L., Sleator D. D., Tarjan R. E. A locally adaptive data compression scheme// Commun. ACM. 1986. V. 29, N 2. P. 320–330.

[45] Burrows M., Wheeler D. J. A block-sorting lossless data compression algorithm// Tech. Rep. Digital System Research Center. Palo Alto, CA, USA. 1994. N 124.

[46] Capocelli R. M., De Santis A. A. New bounds on the redundancy of Huffman codes// IEEE Trans. Inform. Theory. 1991. V. 37, N 4. P. 1095–1104.

[47] Capocelli R. M., De Santis A. A., Cargano L., Vaccaro U. On the construction of statistically synchronizable codes// IEEE Trans. Inform. Theory. 1992. V. 38, N 2. P. 407–414.

[48] Capocelli R. M., Cargano L., Vaccaro U. A fast algorithm for the unique decipherability of multivalued encodings// Theor. Comput. Sci. 1994. V. 134. P. 63–78.

[49] Clarke B. S., Barron A. R. Information-theoretical asymptotics of Bayes methods// IEEE Trans. Inform. Theory. 1990. V. 36, N 3. P. 453–471.

[50] Cover T. M. Enumerative source encoding// IEEE Trans. Inform. Theory. 1973. V. IT-19, N 1. P. 73–77.

[51] Davisson L. D. Universal noiseless coding// IEEE Trans. Inform. Theory. 1973. V. IT-19, N 6. P. 783–795.

[52] Davisson L. D., Leon-Garcia A. A source matching approach to finding minimax codes// IEEE Trans. Inform. Theory. 1980. V. IT-26, N 2. P. 166–174.

[53] Davisson L. D., McEliece R. J., Purley M. B., Wallace M. S. Efficient universal noiseless source codes// IEEE Trans. Inform. Theory. 1981. V. IT-27, N 2. P. 199–207.

[54] Elias P. Universal codeword sets and representations of integers// IEEE Trans. Inform. Theory. 1975. V. IT-21, N 2. P. 194–203.

[55] Elias P. Interval and recency rank source encoding: two on-line adaptive variable-length schemes// IEEE Trans. Inform. Theory. 1987. V. IT-33, N 1. P. 3–10.

[56] Even S., Rodeh M. Economical encoding of commas between strings// Commun. ACM. 1978. V. 21, N 4. P. 315–317.

[57] Ferguson T. J., Rabinowich J. H. Self-synchronizing Huffman codes// IEEE Trans. Inform. Theory. 1984. V. IT-30, N 4. P. 687–693.

[58] Gallager R. G. Variations on a theme by Huffman// IEEE Trans. Inform. Theory. 1978. V. IT-24, N 6. P. 668–674.

[59] Gallager R. G., Van Voorhis D. C. Optimal source codes for geometrically distributed integer alphabets// IEEE Trans. Inform. Theory. 1975. V. IT-21, N 2. P. 228–230.

[60] Gilbert E. N., Moore E. F. Variable-length binary encoding// Bell Syst. Tech. J.. 1959. V. 38, N 4. P. 933–967. (Русский перевод: Гильберт Э. Н., Мур Э. Ф. Двоичные кодовые системы переменной длины// Кибернитический сб. М.: ИЛ, 1961. Вып. 3, С. 103–141.)

[61] Golomb S. W. Run length encoding// IEEE Trans. Inform. Theory. 1966. V. IT-12, N 4. P. 399–401.

[62] Gunther Ch. G. A universal algorithm for homophonic coding// Advances in Cryptology — EUROCRYPT'88. Berlin: Springer-Verlag, 1988. P. 405–414.(Lecture Notes in Comput. Sci. V. 330)

[63] Gutman M. Fixed-prefix encoding of the integers can be Huffman-optimal// IEEE Trans. Inform. Theory. 1990. V. 36, N 4, P. 936–938.

[64] Huffman D. A. A method for the construction of minimum-redundancy codes// Proc. IRE 1952. V. 40, N 10, P. 1098–1101. (Русский перевод: Хаффмен А. Д. Метод построения кодов с минимальной избыточностью// Кибернитический сб. М.: ИЛ, 1961. Вып. 3, С. 79–87.)

[65] Jendal H. N., Kuhn Y. J. B., Massey J. L. An information-theoretic treatment of homophonic substitution// Advances in Cryptology — EUROCRYPT'89. Berlin: Springer-Verlag, 1990. P. 382–394.(Lecture Notes in Comput. Sci. V. 434)

[66] Knuth D. E. Dumamic Huffman coding// J. Algorithms. 1985. V. 6. P. 163–180.

[67] Krichevsky R. E., Trofimov V. K. The performance of universal encoding// IEEE Trans. Inform. Theory. 1981. V. IT-27, N 2. P. 199–207.

[68] Krichevskii R. E. Laplace's law of succession and universal encoding// IEEE Trans. Inform. Theory. 1998. V. 44, N 1. P. 298–303.

[69] Krichevskiy R. E., Potapov V. N. Encoding of run lengths and pyramid cubic lattices// IEEE Trans. Inform. Theory. 1999. V. 45, N 4. P. 1347–1350.

[70] Lam W.-M., Kulkarni S. R. Synchronizing codewords for binary prefix codes// IEEE Trans. Inform. Theory. 1996. V. 42, N 3. P. 984–987.

[71] Lawrence J. C. New universal coding scheme for the binary memoryless source// IEEE Trans. Inform. Theory. 1977. V. IT-23, N 4. P. 466–472.

[72] Leyng-Yan-Cheong S. K., Cover T. M. Some equivalences between Shannon entropy and Kolmogorov complexity// IEEE Trans. Inform. Theory. 1979. V. IT-25, N 3. P. 331–338.

[73] Lounchard G., Szpankowski W. On the average redundancy rate of the Lempel-Ziv code// IEEE Trans. Inform. Theory. 1997. V. 43, N 1. P. 1–7.

[74] Lynch T. J. Data compression, techniques and applications. Bellmont: Lifetime Learning Publications. 1985.

[75] Mandelbrot B. On recurrent noise limiting coding. Laboratories d'Electronique et de physique appliquces. Paris, France, 1954. (Русский перевод: Мандельброт Б. О рекуррентном кодировании, ограничивающем влияние помех. Теория передачи сообщений, М.: ИЛ, 1957. С. 139–157.)

[76] Mandelbrot B. On the theory of word frequencies and on related markovian models of discourse// The structure of language and its mathematical aspects. Providence, RI: Amer. Math. Soc. 1961. P. 190–219. (Proceeding Symposium on Applied Mathematics V. 12)

[77] Mansteteen D. Tight bounds on the redundancy of Huffman codes// IEEE Trans. Inform. Theory. 1992. V. 38, N 1. P. 144–151.

[78] McMillan B. Two inequalities implied by unique decipterability// IRE Trans. Inform. Theory. 1956. V. IT-2. P. 115–116.

[79] Merhav N., Feder M. Universal prediction// IEEE Trans. Inform. Theory. 1998. V. 44, N 6. P. 2124–2147.

[80] Merhav N., Feder M. A strong version of the redundancy-capacity theorem of universal coding// IEEE Trans. Inform. Theory. 1995. V. 41, N 3. P. 714–722.

[81] Moffat A. M. Word based text compression// Software — Practice and Experience. 1989. V. 19, N 2, P. 185–198.

[82] Montgomery B. L., Abrahams J. Synchronization of binary source codes// IEEE Trans. Inform. Theory. 1986. V. IT-32, N 6. P. 849–854.

[83] Morita H., Kobayashi K. On asymptotic optimality of sliding-window variation of Lempel–Ziv codes// IEEE Trans. Inform. Theory. 1993. V. 39, N 6. P. 1840–1847.

[84] Neuhoff D. L., Shields P. S. Simplistic universal coding// IEEE Trans. Inform. Theory. 1998. V. 44, N 2. P. 778–781.

[85] Plotnick E., Weinberger M. J., Ziv J. Upper bounds on the probability of sequences emitted by finite-state source and on the redundancy of the Lempel-Ziv algorithm// IEEE Trans. Inform. Theory. 1992. V. 38, N 1. P. 66–72.

[86] Rissanen J. Generalized Kraft inequality and arithmetic coding// IBM J. Res. Develop. 1976. V. 20, N 3. P. 198–203.

[87] Rissanen J., Langdon G. Universal modelling and coding// IEEE Trans. Inform. Theory. 1981. V. IT-27, N 1. P. 12–23.

[88] Rissanen J. A universal data compression system// IEEE Trans. Inform. Theory. 1983. V. IT-29, N 5. P. 656–664.

[89] Rissanen J. Universal coding, information, prediction and estimation// IEEE Trans. Inform. Theory. 1984. V. IT-30, N 5. P. 656–664.

[90] Rissanen J. Stochactic complexity and modeling// Ann. Statist. 1986. V. 14, P. 1080–1100.

[91] Rissanen J. Fisher information and stochastic complexity// IEEE Trans. Inform. Theory. 1996. V. 42, N 1. P. 40–47.

[92] Rissanen J. Fast universal coding with context models// IEEE Trans. Inform. Theory. 1999. V. 45, N 4. P. 1065–1071.

[93] Rodeh M. A fast test for unique dicipherability based on suffix trees// IEEE Trans. Inform. Theory. 1982. V. IT-28, P. 648–651.

[94] Ryabko B. Ya. Fast and effective coding of information sources// IEEE Trans. Inform. Theory. 1994. V. 40, N 1. P. 96-99.

[95] Savari S. A. Redundancy of the Lempel-Ziv incremental parsing rule// IEEE Trans. Inform. Theory. 1997. V. 43, N 1. P. 8–16.

[96] Savari S. A. Redundancy of the Lempel-Ziv string matching code// IEEE Trans. Inform. Theory. 1998. V. 44, N 2. P. 787–792.

[97] Savari S. A., Gallager R. G. Generalized Tunstall codes for sources with memory// IEEE Trans. Inform. Theory. 1997. V. 43, N 2. P. 658–668.

[98] Schwartz E. S. A dictionary for minimal redundancy encoding// J. Assoc. Comput. Mach. 1963. V. 10, N 4. P. 413–439.

[99] Schutzenberger M. P. On synchronizing prefix codes// Inform. and Control. 1967. V. 11, N 4. P. 396–401.

[100] Shannon C. E. A mathematical theory of communication// Bell System. Tech. J. 1948. V. 27, pt. I. P. 379–423; pt. II, P. 623–656. (Русский перевод: Шеннон К. Работы по теории информации и кибернетике. М.: Изд-во иностр. лит-ры, 1963.)

[101] Shtarkov Y., Babkin V. Combinatorial encoding for discrete stationary sources. 2-nd Internat. Sympos. Inform. Theory. 1971, Budapest: Akad. Kiado, P. 249–257.

[102] Storer J. A., Szymansky T. G. Data compression via textual substitution// J. Assoc. Comput. Mach. 1982. V. 25, N 4, P. 928–951.

[103] Stout Q. F. Improved prefix encoding of the natural numbers// IEEE Trans. Inform. Theory. 1980. V. IT–26. P. 607–609.

[104] Titchener M. R. The synchronization on variable-length codes// IEEE Trans. Inform. Theory. 1997. V. 43, N 2. P. 683–691.

[105] Tjalkens T. J., Willems F. M. Variable to fixed-length codes for Markov sources// IEEE Trans. Inform. Theory. 1987. V. IT-33, N 2. P. 337–343.

[106] Tunstall B. P. Synthesis of noiseless compression codes: Ph.D. dissertation. Atlanta, GA: Georgia Inst. Technol, 1967.

[107] Vitter J. S. Design and analysis of dunamic Huffman codes// J. Assoc. Comput. Mach. 1987. V. 34, N 4. P. 825–845.

[108] Weber A., Head T. The finest homofonic partition on related code concepts// IEEE Trans. Inform. Theory. 1996. V. 42, N 5. P. 1569–1575.

[109] Weinberger M. J., Rissanen J., Feder M. A universal finite memory source// IEEE Trans. Inform. Theory. 1995. V. 41, N 3. P. 643–652.

[110] Welch. T. A. A technique for high-performance data compression// IEEE Computers. 1984. V. 17, N 6. P. 8–19.

[111] Willems F. M. J.. The context-tree weighting method: extensions// IEEE Trans. Inform. Theory. 1998. V. 44, N 2. P. 792–798.

[112] Willems F. M. J., Shtarkov Y. M., Tjalkens T. J. The context-tree weighting method: basic properties// IEEE Trans. Inform. Theory. 1995. V. 41, N 3. P. 653–664.

[113] Willems F. M. J., Shtarkov Y. M., Tjalkens T. J. Context weighting for general finite-context sources// IEEE Trans. Inform. Theory. 1996. V. 42, N 5. P. 1514–1520.

[114] Witten I. H., Neal R. M., Cleary J. G. Arithmetic coding for data compression// Commun. ACM. 1987. V. 30, N 6. P. 520–540.

[115] Wyner A. D., Ziv J. Some asymptotic properties of the entropy of a stationary ergodic data source with applications to data compression// IEEE Trans. Inform. Theory. 1989. V. 35, N 6. P. 1250–1258.

[116] Wyner A. D., Wyner A. J. Improved redundancy of a version of the Lempel-Ziv algorithm// IEEE Trans. Inform. Theory. 1995. V. 41, N 3. P. 723–731.

[117] Wyner A. D., Ziv J., Wyner A. J. On the role of pattern matching in information theory// IEEE Trans. Inform. Theory. 1998. V. 44, N 6. P. 2045–2056.

[118] Wyner A. J. The redundancy and distribution of phrase lengths of the fixed-database Lempel-Ziv algorithm// IEEE Trans. Inform. Theory. 1997. V. 43, N 5. P. 1452–1464.

[119] Xie Q., Barron A. R. Minimax redundancy of the class of memoryless source// IEEE Trans. Inform. Theory. 1997. V. 43, N 2. P. 648–657.

[120] Yokoo H. Improved variations relating the Ziv–Lempel and Welch-type algorithms for sequential data compression// IEEE Trans. Inform. Theory. 1992. V. 38, N 1. P. 78–81.

[121] Zipf G. K. The psychobiology of language. Boston: Houghtton–Mifflin, 1935.

[122] Ziv J., Lempel A. A universal algorithm for sequential data compression// IEEE Trans. Inform. Theory. 1977. V. IT-23, N 3. P. 337–343.

[123] Ziv J., Lempel A. Compression of individual sequences via variable–length coding// IEEE Trans. Inform. Theory. 1978. V. IT-24, N 5. P. 530–536.