ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Июль—декабрь 2007. Серия 2. Том 14, № 2. 3–13

УДК 519.722

ИСПОЛЬЗОВАНИЕ ЗАКОНА ЦИПФА ДЛЯ СЖАТИЯ ТЕКСТОВ

М. П. Бакулина

Рассматривается одна из важных задач теории информации — задача сжатия данных, в частности, текстов на естественных языках, с сохранением возможности их однозначного восстановления (декодирования). Предлагается один из способов решения этой задачи — построение кодов, базирующихся на законе Ципфа. В отличие от универсальных методов такое построение использует знания о статистической структуре источника сообщений. Рассматриваются алгоритмы двухпроходной и однопроходной схем кодирования и оценивается эффективность их сжатия.

Введение

Сжатие данных является одной из важных задач теории информации. Это связано с тем, что задача компактного представления данных с сохранением возможности их однозначного восстановления (декодирования) возникает достаточно часто на практике. Так, предварительное сжатие позволяет существенно улучшить характеристики системы связи. Сжатие применяется при хранении информации, её передаче со спутников и в других приложениях, причём актуальность проблемы сжатия возрастает в связи с увеличением объёмов передаваемой и хранимой информации.

В настоящее время широко известны и находят практическое применение многие алгоритмы компактного представления данных, которые дают сжатие лучше, чем стандартные архиваторы. Кроме алгоритмов, использующих побуквенное кодирование, примером которого является известный код Хаффмана [4], известны также методы словарного сжатия данных [5, 10, 11]. В таких методах для имеющегося сообщения, например, текста на естественном языке, составляется частотный словарь, представляющий собой список всех слов, используемых в тексте, с указанием частоты их появления. Тогда при кодировании сообщения каждому слову приписывается код, длина которого тем меньше, чем выше частота его встречаемости.

Одним из основных подходов к задаче эффективного кодирования данных, в частности, текстов на естественных языках, может служить построение кодов, базирующихся на общих законах лингвистики, например, на законе Ципфа [13]. В отличие от универсальных методов этот закон использует знания о статистической структуре источника сообщений. На основе закона Ципфа в данной статье предлагаются новые алгоритмы кодирования текстов. Эти методы обладают хорошей степенью сжатия и высокой скоростью кодирования. Степень (коэффициент) сжатия текста здесь определяется отношением объёма текста, полученного в результате кодирования, к его первоначальному объёму.

В разделе 2 рассматривается двухпроходная схема сжатия на основе двух способов кодирования слов (равномерного и неравномерного) и устанавливается оценка отношения степеней сжатия, полученных при таких способах кодирования. Наряду с теоретическим результатом был проведён ряд экспериментов, в которых использовались различные тексты на естественных языках с разными объёмами словарей и подсчитывалась степень их сжатия. Сравнение показало, что результаты экспериментов хорошо согласуются с результатами, полученными аналитически. В разделе 3 рассматриваются алгоритмы однопроходной схемы сжатия с известным (неизвестным) объёмом словаря. В разделе 4 приводятся наиболее известные эффективные методы кодирования, а также сравнение результатов сжатия, полученных этими методами, с результатами, полученными алгоритмами, основанными на законе Ципфа.

1. Закон Ципфа

Этот закон был установлен Г. Ципфом на основе анализа текстов на различных языках и состоит в следующем. Пусть имеется текст на каком-либо естественном языке, L — объём словаря этого текста. Расположим слова в словаре в порядке убывания частот употребления и пронумеруем их от 1 до L. Тогда зависимость между частотой и номером слова может быть описана формулой

$$f_i = \frac{k}{i^\alpha},$$

где f_i – частота i-го слова, k — величина, зависящая от объёма словаря, α — константа. Эта зависимость и получила название закона Ципфа. В дальнейшем закон подтверждён многими исследователями для всех европейских языков [1]. В частности, Б. Мандельброт [8] показал, что закон Ципфа будет выполняться, если пробел между словами рассматривать как случайный символ. Он же получил это распределение, исходя

из предположения, что эволюционный процесс выбора длин слов может быть описан как случайное блуждание [9].

Разделим обе части данного равенства на N, где N — общее число слов в тексте. Тогда закон Ципфа можно записать в виде

$$p_i = \frac{c}{i^\alpha},$$

где p_i — вероятность слова, $c = k/N$. Известно, что для слов естественного языка $\alpha \approx 1$ и закон Ципфа задаётся формулой $p_i = c/i$, которой мы и будем пользоваться в дальнейшем изложении. Отметим также, что так как $\ln p_i = \ln c - \ln i$, то зависимость между частотой и номером слова в словаре в логарифмических координатах практически близка к линейной, т. е. может быть представлена прямой на графике (с осями $\ln i$ и $\ln p_i$).

2. Равномерное и неравномерное по выходу кодирование слов

Пусть D — объём словаря некоторого текста на естественном языке. Расположим все слова в словаре по убыванию их частот и рассмотрим равномерное (неравномерное) по выходу кодирование слов с использованием закона Ципфа. Алгоритм кодирования, при котором сначала составляется частотный словарь, а потом производится кодирование слов в словаре на основе имеющейся статистики, назовём двухпроходной схемой сжатия.

При кодировании Шеннона каждому слову ставится в соответствие кодовое слово длины $\lceil -\log p_i \rceil$, где p_i — вероятность i-го слова. При равномерном кодировании длина кодового слова равна $\log D$. Тогда длина текста, полученного при кодировании Шеннона, равна

$$B_1 = N \cdot \sum_{i=1}^{D} p_i \lceil -\log p_i \rceil, \tag{1}$$

где N — общее число слов в тексте. При равномерном кодировании длина текста равна

$$B_2 = N \cdot \lceil \log D \rceil. \tag{2}$$

Наша задача состоит в выяснении того, можно ли получить существенный выигрыш в сжатии при неравномерном кодировании по сравнению с равномерным. Под коэффициентом (степенью) сжатия текста мы понимаем отношение объёма текста (в битах), полученного в результате кодирования, к первоначальному объёму текста.

Теорема. *Пусть имеется текст с объёмом словаря D, в котором распределение слов по частоте подчинено закону Ципфа, n_1, n_2 — коэффициенты сжатия, полученные при неравномерном и равномерном по выходу кодировании слов соответственно. Тогда* $\dfrac{n_1}{n_2} \to \dfrac{1}{2}$ *при* $D \to \infty$.

Доказательство. Оценим объём текста B_1, полученный при неравномерном кодировании. Используя (1) и учитывая, что

$$-\log p_i \leqslant \lceil -\log p_i \rceil < -\log p_i + 1,$$

получаем

$$N \cdot \sum_{i=1}^{D} p_i(-\log p_i) \leqslant B_1 < N \cdot \sum_{i=1}^{D} p_i(-\log p_i + 1). \tag{3}$$

Согласно закону Ципфа

$$p_i = \frac{k}{i}, \tag{4}$$

где k — константа, зависящая от D. Для оценки этой константы и дальнейших преобразований воспользуемся известными оценками конечных сумм (см. [3]):

$$\ln(D+1) < \sum_{i=1}^{D} \frac{1}{i} < \ln(D+1) + c_1, \tag{5}$$

$$\frac{\ln^2(D+1)}{2} + c_2' < \sum_{i=1}^{D} \frac{\ln i}{i} < \frac{\ln^2(D+1)}{2} + c_2, \tag{6}$$

где $c_1 = 0{,}577\ldots$ — постоянная Эйлера, $c_2' = -0{,}3$, $c_2 = -0{,}105\ldots$ (численное значение c_2 можно получить из ряда Эйлера–Маклорена [3]). Так как $k \cdot \sum\limits_{i=1}^{D} \frac{1}{i} = 1$, то

$$\frac{1}{\ln(D+1) + c_1} < k < \frac{1}{\ln(D+1)}. \tag{7}$$

Применяя формулы (3)–(7) и известное неравенство $\ln(1+x) < x$ при

$x > -1$, получаем

$$B_1 < N \cdot \Big(\sum_{i=1}^{D} \frac{k}{i}(\log i - \log k + 1)\Big) = \frac{N \cdot k}{\ln 2} \cdot \Big(\sum_{i=1}^{D} \frac{\ln i}{i} + (\ln 2 - \ln k) \cdot \sum_{i=1}^{D} \frac{1}{i}\Big)$$
$$< \frac{N}{\ln 2 \cdot \ln(D+1)} \cdot \Big(\frac{\ln^2(D+1)}{2} + c_2 + (\ln 2 + \ln(\ln(D+1) + c_1)) \cdot (\ln(D+1) + c_1)\Big) < \frac{N}{\ln 2 \cdot \ln(D+1)} \Big(\frac{\ln^2(D+1)}{2} + c_2 + \Big(\ln 2 + \ln\ln(D+1) + \frac{c_1}{\ln(D+1)}\Big) \times (\ln(D+1) + c_1)\Big) = \frac{N}{\ln 2 \cdot \ln(D+1)} \cdot \Big(\frac{\ln^2(D+1)}{2} + c_1 \cdot (1 + \ln 2) + c_2 + \ln\ln(D+1) \cdot (\ln(D+1) + \ln 2) + c_1 \cdot \ln\ln(D+1) + \frac{c_1^2}{\ln(D+1)}\Big). \quad (8)$$

Учитывая (2), получаем оценки для длины текста B_2 при равномерном кодировании:

$$N \cdot \log D \leqslant B_2 < N \cdot (\log D + 1). \quad (9)$$

Если $n_1 = B_1/A$ и $n_2 = B_2/A$ — коэффициенты сжатия текста, полученные при неравномерном и равномерном кодировании соответственно, где A — первоначальная длина текста, то, используя формулы (7) и (8), получаем верхнюю оценку для отношения n_1/n_2:

$$\frac{n_1}{n_2} = \frac{B_1}{B_2} < \frac{\ln(D+1)}{2\ln D} + \frac{c_1 \cdot (1 + \ln 2) + c_2}{\ln(D+1) \cdot \ln D} + \frac{\ln\ln(D+1)}{\ln D} + \frac{\ln\ln(D+1) \cdot (\ln 2 + c_1)}{\ln(D+1) \cdot \ln D}. \quad (10)$$

При $D \to \infty$ правая часть выражения (10) стремится к $1/2$. Аналогично, используя (3)–(7), имеем

$$B_1 \leqslant N \sum_{i=1}^{D} p_i(-\log p_i) = N \sum_{i=1}^{D} \frac{k}{i}(\log i - \log k)$$
$$= \frac{Nk}{\ln 2}\Big(\sum_{i=1}^{D} \frac{\ln i}{i} - \ln k \sum_{i=1}^{D} \frac{1}{i}\Big) > \frac{N}{\ln 2 \cdot (\ln(D+1) + c_1)} \cdot \Big(\frac{\ln^2(D+1)}{2} + c_2' + \ln\ln(D+1) \cdot \ln(D+1)\Big). \quad (11)$$

Учитывая (9), из формулы (11) получаем нижнюю оценку для отношения n_1/n_2:

$$\frac{n_1}{n_2} > \frac{1}{2} - \frac{1}{\ln(D+1)+1} + \frac{c_2'}{(\ln(D+1)+1)^2} + \frac{\ln\ln(D+1)\cdot\ln(D+1)}{(\ln(D+1)+1)^2}. \quad (12)$$

При $D \to \infty$ правая часть выражения (12) стремится к 1/2. Следовательно, $n_1/n_2 \to 1/2$ при $D \to \infty$. Теорема доказана.

Из теоремы следует, что при неравномерном кодировании слов для текстов на естественных языках можно добиться улучшения сжатия не более чем в два раза по сравнению с равномерным кодированием.

Полученный теоретический результат был подтверждён экспериментально. Для эксперимента брались различные тексты на русском и английском языках с разными объёмами словарей и подсчитывались коэффициенты их сжатия при равномерном и неравномерном кодировании. В таблице 1 приведены коэффициенты сжатия текстов n_1, n_2, выраженные в процентах (экспериментальные результаты приведены в скобках), объёмы словарей, а также отношение n_1/n_2, полученное теоретически и экспериментально.

Таблица 1

№	Название	D	n_1 (%)	n_2 (%)	n_1/n_2 теор.	n_1/n_2 эксп.
1	article	1132	17,4(17,5)	21,7(21,8)	0,802	0,803
2	book	2100	17,9(18,1)	23,2 (23,3)	0,772	0,777
3	journal	2480	17,2(17,3)	22,5(22,6)	0,764	0,765
4	document	3700	17,5(17,7)	23,8(24)	0,735	0,738
5	works	21197	16,6(16,7)	25,4(25,5)	0,654	0,655

Здесь для эксперимента были взяты:

article — статья в журнале (67739 байт);

book — монография (349270 байт);

journal — Сибирский математический журнал (307930 байт);

document — документация (962186 байт);

works — собрание сочинений А. С. Пушкина (5410342 байт).

Как видно из таблицы 1, с ростом объёма словаря отношение коэффициентов сжатия уменьшается, причём значительный выигрыш в сжатии наблюдается при большом объёме словаря. Отметим также, что на практике в связи с ограниченностью текста (и словаря) экспериментальный результат лишь приближается к теоретическому пределу 1/2, не достигая его (например, для гипертекста works размером 5410 кбайт с достаточно большим объёмом словаря 21197 слов отношение n_1/n_2 равно 0,654).

3. Однопроходная схема сжатия

Теперь рассмотрим однопроходную схему сжатия, основанную на законе Ципфа. В отличие от двухпроходной схемы, где предполагается, что на начальном этапе кодирования словарь уже составлен, в однопроходном варианте словарь наполняется по мере прохождения текста.

Сначала рассмотрим случай, когда на начальном шаге объём словаря известен. Пусть D — общее число слов в словаре. Кодирование слов будем осуществлять следующим образом. Составим частотный словарь для текста, прочитанного до определённого слова (это слово предстоит закодировать). Если оно не встречается в текущем словаре (т. е. новое), то записываем его в словарь и кодируем побуквенно. Если же слово уже встречалось, то кодируем его в соответствии с законом Ципфа, при этом длина кода равна $l_i = \lceil -\log \frac{c(D)}{i} \rceil$ где $c(D) \approx \frac{1}{\ln D}$, а i — номер слова в словаре. После прочтения очередного слова в тексте и занесения его в словарь все слова в текущем словаре переупорядочиваем заново и располагаем их в порядке убывания частот. Перед кодом каждого нового слова будем вставлять специальное кодовое слово (флаг), длина которого равна $\lceil -\log(1 - \sum_{i=1}^{j} \frac{c(D)}{i}) \rceil$, где j — число слов в текущем словаре. При декодировании сообщения на приёмном конце легко определяется флаг (получатель его знает) и идущая за ним кодовая последовательность.

Теперь рассмотрим случай, когда на начальном шаге объём словаря неизвестен. Алгоритм кодирования строится так же, как и при известном объёме. Отличие заключается лишь в том, что на каждом шаге неизвестная величина $c(D)$ оценивается по текущему словарю. Это означает, что если D_j — число слов в словаре, составленном для текста, прочитанного до j-го слова, то полагаем $c(D_j) \approx \frac{1}{\ln D_j}$.

В таблице 2 приведены результаты сжатия, полученные при однопроходном кодировании с известным и неизвестным объёмом словаря для тех же текстов, что и в таблице 1, т. е. article, book, journal, document и works.

Т а б л и ц а 2

№	Название	V текста (байт)	k с изв. сл. (%)	k с неизв. сл. (%)
1	article	67739	24,7	25,1
2	book	349270	24,9	25,7
3	journal	307930	24,2	25,3
4	document	962186	25,4	26,2
5	works	5410342	25,8	26,5

Отметим, что наилучшее сжатие даёт двухпроходный вариант (см. таблицу 1), а из таблицы 2 следует, что в однопроходной схеме сжатие при известном объёме словаря лучше сжатия, полученного при неизвестном объёме словаря.

4. Основные алгоритмы сжатия, сравнение результатов

Для оценки эффективности предложенной однопроходной схемы кодирования, основанной на законе Ципфа, было проведено сравнение результатов сжатия с результатами, даваемыми другими известными алгоритмами.

Одна из известных схем сжатия, основанная на сведении к минимуму избыточности, — алгоритм Лемпеля–Зива [14, 15] (именуемый LZ-алгоритмом) и его модификации (например, модификация Велча [12]). В LZ77-алгоритме [14] кодируемое слово разделяется по определенному правилу на подслова, кодами которых являются пары чисел. Схема кодирования LZ78 [15] отличается от предыдущей тем, что на каждом шаге выбирается наиболее длинное начало остатка, которое совпадает с некоторым уже выделенным подсловом, и к нему добавляется ещё одна буква. Отметим, что LZ78-алгоритм можно рассматривать как кодирование с динамичным словарем. Словарь можно произвольным образом нумеровать и удалять из него слова, все продолжения которых уже имеются в словаре.

Наиболее распространенные и эффективные методы кодирования приведены также в [5]. Остановимся на некоторых из них. Хорошее сжатие среди описанных в [5] алгоритмов даёт метод PPM, предложенный Дж. Клиэри и И. Виттеном. В этом методе в процессе кодирования вместо безусловных вероятностей букв оцениваются их условные вероятности при известном «контексте», т. е. при известных предшествующих буквах. В PPM используются контексты различной длины, зависящей от предшествующей закодированной последовательности данных. Основная проблема PPM — это оценка вероятности, которую имеют ещё не появившиеся в контексте символы. Для решения этой проблемы в [5] предложены варианты PPM — алгоритмы PPMA и PPMB. В них используются априорные методы, основанные на предположениях о природе сжимаемых данных. Ещё одна усовершенствованная версия PPM — алгоритм PPMC, разработанный в [10]. Данный метод позволяет улучшить сжатие и увеличить скорость кодирования. Представляют интерес описанные в [5] метод статистического моделирования DMC, алгоритм WORD, где символами являются слова, и схема MTF. Алгоритм MTF разработан Б. Я. Рябко [2] и назван методом стопки книг. Впоследствии

метод стопки книг был переоткрыт П. Элайесом [7] под названием «кодирование по степени новизны» и Дж. Бентли с соавторами [6] под названием «двигай вверх». В MTF-алгоритме текст рассматривается как чередующаяся последовательность слов и символов, не являющихся словами. Каждое слово в тексте после его очередного появления удаляется с текущей позиции и перемещается вперед. В результате слова́, которые в тексте появляются часто, оказываются впереди слов, встречающихся редко. При этом словам, расположенным впереди, присваиваются более короткие коды. Если слово (символ) появляется впервые, то оно (он) кодируется побуквенно, иначе кодирование производится с учётом частоты появления слова в тексте. Посылаемое дополнительно кодовое слово (флаг) позволяет определить, является ли приходящее слово уже встречавшимся или новым.

Для сравнения результатов автором были практически реализованы некоторые наиболее эффективные алгоритмы из [5], в частности, схемы PPMC, WORD, DMC, MTF и новый описанный выше алгоритм ZIPF, основанный на законе Ципфа, (в частности, однопроходная схема кодирования с неизвестным объёмом словаря), а также известный алгоритм оптимального кодирования Хаффмана (HUF) и алгоритм Лемпеля–Зива LZ78. Экспериментальные результаты сжатия, даваемые этими алгоритмами для различных тестовых файлов, были получены на IBM PC с процессором Pentium и оперативной памятью 256 Мб под операционной системой Windows 98. В таблице 3 приведены результаты сжатия в наиболее удобной и чаще употребляемой в литературе единице измерения степени сжатия бит/символ.

Т а б л и ц а 3

№	Название	Сжатие (бит/символ)						
		HUF	PPMC	WORD	DMC	MTF	LZ	**ZIPF**
1	article	2,09	2,46	2,58	2,67	2,72	2,84	**2,25**
2	book	2,02	2,32	2,75	2,55	2,90	3,12	**2,18**
3	journal	2,03	2,48	2,54	2,51	2,66	2,76	**2,16**
4	document	2,05	2,36	2,39	2,82	2,71	2,78	**2,23**
5	works	2,01	2,26	2,36	2,55	2,49	2,60	**2,12**

Кроме степени сжатия важной характеристикой метода является скорость (время) кодирования. Для рассмотренных выше методов в таблице 4 приведены экспериментальные результаты времени сжатия. В качестве результата принималось усредненное общее время работы программы по 10 запускам (использовался таймер в 1000 Гц).

Из таблицы 3 видно, что коэффициент сжатия, полученный с помощью метода, основанного на законе Ципфа, больше только коэффициента сжатия с помощью оптимального кода Хаффмана, но меньше коэффициентов сжатия, полученных другими рассмотренными методами. Сравнение же алгоритмов по времени (таблица 4) показывает, что код Хаффмана имеет наибольшее время кодирования, тогда как предложенный метод ZIPF имеет меньшее время кодирования, большим времени, полученным алгоритмом Лемпеля–Зива, но имеющим меньшим коэффициентом сжатия. Таким образом, результаты сравнения показывают хорошую степень и скорость сжатия метода кодирования на основе закона Ципфа, что подтверждает эффективность построенного алгоритма.

Таблица 4

№	Название	Время (с)						
		HUF	PPMC	WORD	DMC	MTF	LZ	**ZIPF**
1	article	2,95	1,61	1,44	1,15	1,07	0,83	**0,97**
2	book	9,78	8,32	7,16	6,48	5,69	5,11	**5,42**
3	journal	8,81	7,45	6,33	5,96	5,18	4,76	**4,90**
4	document	24,14	22,83	20,84	19,17	19,09	18,51	**18,85**
5	works	75,22	73,05	70,62	68,94	68,13	66,48	**67,01**

ЛИТЕРАТУРА

1. **Пиотровский Р. Г.** Текст, машина, человек. Л.: Наука, 1975.
2. **Рябко Б. Я.** Эффективный метод кодирования источников информации, использующий алгоритм быстрого умножения // Проблемы передачи информации. 1995. Т. 31, вып. 1. С. 3–12.
3. **Фихтенгольц Г. М.** Курс дифференциального и интегрального исчисления. Т. 2. М.: Физматлит, 2001.
4. **Хаффман Д. А.** Метод построения кодов с минимальной избыточностью // Кибернетический сборник. Вып. 3. М.: ИЛ, 1961. С. 79–87.
5. **Bell T. C., Cleary J. H., Witten I. H.** Text compression. Prentice Hall. Englewood Cliffs, 1990.
6. **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.
7. **Elias P.** Interval and recency rank source encoding: two on-line adaptive variable-length schemes // IEEE Trans. Inform. Theory. 1987. V. 33, N 1. P. 3–10.

8. Mandelbrot B. On recurrent noise limiting coding. Lab. d'Electronique et de physique appliquces. Paris, 1954. (Рус. пер.: **Мандельброт Б.** О рекуррентном кодировании, ограничивающем влияние помех // Теория передачи сообщений. М.: ИЛ, 1957. С. 139–157.)

9. 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.)

10. Moffat A. M. Word based text compression // Software-Practice and Experience. 1989. V. 19, N 2. P. 185–198.

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

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

13. Zipf G. K. Human behavior and the principle of least effort. Cambridge: Addison Wesley, 1949.

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

15. 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.

Адрес автора:

Институт вычислительной математики и
математической геофизики СО РАН
пр. Академика Лаврентьева, 6
630090 Новосибирск,
Россия.
E-mail: marina@rav.sscc.ru

Статья поступила
17 мая 2007 г.
Переработанный вариант —
29 октября 2007 г.