ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ

УДК 621.391.1

Р. Н. БОГАТОВ

Омский государственный технический университет

МЕТОДЫ МАКСИМАЛЬНОГО НЕИСКАЖАЮЩЕГО СЖАТИЯ

В статье приводится обзор методов сжатия с точки зрения достижения наибольшей степени сжатия «типичных» данных. Особое внимание уделено контекстному моделированию (КМ) и методам, использующим PPM (Prediction by Partial Match – предсказание по частичному совпадению), которые достигают наибольшей известной степени сжатия. Рассмотрены наиболее существенные недостатки КМ. Предложены три возможных пути повышения эффективности КМ: ассоциативное КМ, использование удаленных контекстов и двустороннее КМ.

Сжатием данных называют обработку данных с целью получения их более краткого представления. *Неискажающее сжатие* подразумевает возможность разжатия *без потери информации*, что означает идентичность представления разжатых и исходных данных. Успех сжатия зависит от природы и количества данных, наличия априорных знаний об их природе, характеристик алгоритма сжатия, использованных вычислительных ресурсов и других факторов. Основными показателями эффективности сжатия являются:

– степень сжатия, равная отношению длин представлений сжатых и исходных данных;

– время сжатия;

– объем оперативной памяти, необходимой для осуществления сжатия;

– время разжатия;

– объем оперативной памяти, необходимой для разжатия.

Под максимальным сжатием данных подразумевают универсальный алгоритм, обеспечивающий наивысшую среднюю степень сжатия конечного подмножества данных определенного класса по сравнению с другими методами. При этом максимальная возможная средняя степень сжатия, которая может быть достигнута, остается неизвестной по причине невычислимости колмогоровской сложности. Целесообразность разработки новых методов сжатия обуславливается стабильностью динамики роста максимальной достигнутой степени сжатия тестовых наборов данных.

Задача сжатия может быть сформулирована только относительно определенного класса данных. Под *типичными данными* будем понимать информацию,

представленную в структурах современных информационных систем и непосредственно используемую на практике. Общее свойство таких данных заключается в наличии закономерностей представления, обусловленных их техническим происхождением, т.е. наличии избыточности, позволяющей осуществить их сжатие. Примером могут послужить текстовая информация в любом виде, базы данных, файлы ресурсов приложений, исполнимые файлы и т.д.

При *многопроходном* сжатии производится предварительный анализ всего объема данных с целью выявления их внутренней структуры, осуществления возможной декомпозиции, выбора модели для сжатия и настройки параметров модели. На последнем проходе на основе собранной статистики осуществляется сжатие данных с предварительным кодированием информации, необходимой для осуществления разжатия. *Однопроходные* методы сжатия не используют предварительного анализа всего объема данных. Сжатие осуществляется из априорного предположения о природе данных, и на основе уже обработанной части подстраиваются параметры модели (производится *адаптация*). В 1981 году Й. Й. Риссанен и Г. Г. Лангдон предложили называть такой подход *универсальным кодированием* и выделять в нем две функциональные составляющие, *моделирование* и *кодирование*, осуществляемые последовательно для сжатия каждого сообщения источника [1]. *Модель*, в конечном счете, является способом оценки вероятностей возможных значений следующего ожидаемого сообщения источника. Задачей *кодера* является, используя полученные от модели вероятности, построить для очередного сообщения сжатый код, максимально приближенный по числу бит к энтропии этого сообщения. После кодирования модель производит необходимую адаптацию, оценивает вероятности возможных значений следующего сообщения, новое сообщение вместе с априорными оценками модели передается кодеру и кодируется, после чего цикл повторяется для каждого кодируемого сообщения. Такая последовательность обновления модели позволяет осуществить аналогичные действия при разжатии: на основании текущих оценок модели *декодер* производит восстановление значения сжатого сообщения, модель адаптируется и подготавливает декодер для принятия следующего сообщения. Только небольшая часть современных методов сжатия, обеспечивающих высокую степень сжатия типичных данных, не использует явного моделирования как функционально отделимого элемента. К ним относятся, в частности, методы, использующие BWT (см. далее в статье).

Для достижения максимальной степени сжатия требуется построение модели, адекватной источнику входных данных. Выбор способа кодирования промоделированных данных влияет на степень сжатия в меньшей степени. Так, например, в связке со словарным моделированием для повышения быстродействия часто используется равномерное кодирование (схемы LZ77, LZ78, LZW, LZSS и др.). Разница в степени сжатия по сравнению с методами, использующими оптимизированное кодирование (LZH, LZFG, LZC и др.), является гораздо меньшей величиной, чем в случае использования различных типов моделирования и одного метода кодирования [2, 3].

Словарные модели зарекомендовали себя в качестве универсальных для сжатия типичных данных, часто обладающих Марковскими свойствами. Для достижения максимальной степени сжатия используют методы *контекстного моделирования* (КМ), обеспечивающие всегда лучшее сжатие по сравнению со словарными методами (при сжатии типичных данных с Марковскими свойствами), но требующие больших вычислительных затрат. Поскольку предметом настоящего исследования являются методы достижения максимальной степени сжатия, основное внимание будет уделено лидирующим по этому критерию методам КМ и возможным способам повышения их эффективности.

Большинство методов КМ являются *стохастическими*, оценивающими вероятности значений следующего элемента данных на основе статистики по обработанной части и текущего контекста. Исключение составляют методы, использующие *преобразование Барроуза-Уиллера* (BWT, сокр. от англ. Burrows-Wheeler Transform), которое избавляет данные от контекстной избыточности с помощью сортировки. Среди стохастических методов могут быть выделены использующие *явное взвешивание* оценок текущих (активных) контекстов (*взвешивание контекстного дерева* или CWT от англ. Context Tree Weighting, *динамическое Марковское сжатие* или DMC от англ. Dynamic Markov Compression и др.) и методы, использующие *неявное взвешивание* с помощью техники *предсказания по частичному совпадению* (PPM, сокр. от англ. Prediction by Partial Match), обеспечивающие наибольшую известную степень сжатия типичных данных и получившие наибольшее распространение.

Суть классического PPM, предложенного в 1984г. Дж. Г. Клэари (Cleary) и И. Х. Уиттеном (Witten) [4], заключается в следующем. Пусть последними символами источника были $...s_{k-3}s_{k-2}s_{k-1}$ и имеется M контекстных моделей, предсказывающих значение следующего символа s_k на основе статистики, накопленной по контекстам разной длины. Контекстная модель порядка m содержит словарь контекстов (цепочек символов) длины m, встречавшихся ранее более одного раза, и для каждого контекста — счетчики символов, которые ранее встречались следующими за ним. Для данного случая m-тая модель обеспечит статистику по цепочкам вида $s_{k-m}...s_{k-2}s_{k-1}x$, встречавшимся ранее, отличающимися значением x. Кроме M контекстных моделей используются две условных модели нулевого и минус первого порядка. Модель нулевого порядка предсказывает значение s_k на основе накопленных частот встретившихся символов; модель минус первого порядка полагает все возможные значения s_k равновероятными.

Для оценки значения s_k выбирается одна из $M+2$ моделей, обладающая статистикой по текущему контексту, и предпринимается попытка закодировать s_k на основе ее распределения вероятностей. В классическом варианте модели просматриваются, начиная со старших порядков. Если выбранная модель не может закодировать s_k (такое значение в данном контексте не ожидалось), то кодируется *код ухода*, означающий необходимость использования другой модели. Если все M контекстных моделей и модель нулевого порядка выдадут код ухода, s_k кодируется равномерным кодом в модели минус первого порядка.

Оценка вероятности ухода (ОВУ) является одной из ключевых проблем сжатия с использованием PPM. Большинство известных методов априорной ОВУ (методы A, B [4], C, D [5], P, X, XC [3]) используют только три величины: C — частоту появления данного контекста ранее; S — кол-во различных символов, появлявшихся ранее после данного контекста и $S^{(i)}$ — кол-во различных символов, появлявшихся ранее после данного контекста только i раз. Бо́льшую точность дают адаптивные методы ОВУ, использующие повторную оценку ухода (SEE, сокр. от англ. Secondary Es-

cape Estimation), основанную на текущей статистике уходов. Адаптивные методы Z [6] Ч. Блума (Bloom) с вариациями, SEE-d1 и SEE-d2 Д. А. Шкарина [7] позволяют достичь наибольшей степени сжатия и используются в современных архиваторах.

Для повышения степени сжатия PPM также применяют следующие дополнительные техники:

– учет статистики по контекстам неограниченного порядка (PPM* [8]);

– частичное обновление или "исключение при обновлении" счетчиков (update exclusion) вместо полного обновления (full updates), а также увеличение шага прироста счетчиков;

– просмотр моделей на основе иного критерия, нежели величины порядка модели, для чего применяется оценка локального порядка модели (LOE, от англ. Local Order Estimation) на основе вероятности наиболее вероятного в каждой модели символа (MPS-P, от англ. Most Probable Symbols's Probability) и счетчиков уходов из этих моделей [6];

– наследование информации (information inheritance) дочерними контекстными моделями (большего порядка), из которых был выполнен уход, у родительской модели, закодировавшей символ [7];

– технику масштабирования новизны (recency scaling), заключающуюся в искусственном завышении счетчика последнего встретившегося в каждом контексте символа [3];

– вторичную оценку символов (SSE, от англ. Secondary Symbol Estimation) с помощью явного взвешивания статистик родительских контекстов [7].

По результатам различных тестов (www.maximumcompression.com, http://www.compression.ru/ybs и др.) наибольшая возможная на сегодняшний день степень сжатия типичных данных достигается при использовании PPMII (PPM с наследованием информации; от англ. PPM with Information Inheritance) – комплексной модификации алгоритма PPM*, предложенной Д. А. Шкариным в 2001г., используемой в большинстве современных лидирующих архиваторов.

В связке с КМ чаще всего используется *арифметическое кодирование*, обладающее наименьшей избыточностью (см., например, [9]). Избыточность современных алгоритмов кодирования составляет крайне малую долю от избыточности, внесенной на этапе моделирования, и более не является предметом исследований, имеющих целью достижение большей степени сжатия.

Подробное описание вышеупомянутых техник, а также сравнительный анализ современных методов сжатия могут быть найдены в работе Д. Ватолина с соавторами [3], в обзорах В. Н. Потапова [10] и К. Ю. Балашова [2], диссертациях А. В. Кадача [11], С. Бантон [12] и П. Г. Ховарда [5]. Оставшаяся часть статьи посвящена обзору дополнительных механизмов повышения степени сжатия типичных данных, а также анализу недостатков и возможных способов развития КМ.

Для некоторых классов данных (например, различных текстов), а также для определенных распространенных форматов данных (например, исполнимых двоичных файлов, документов Microsoft Office и др.) применяется предобработка с помощью специальных *преобразующих фильтров*, представляющих входные данные в более «понятной» для моделирования форме. Например, при универсальном кодировании текстов с помощью КМ, оказываются полезными следующие преобразования:

– отделение символов пунктуации дополнительными пробелами (space stuffing);

– замена прописных букв строчными (capital conversion);

– специальная обработка кода конца строки (EOL removing; EOL от англ. End Of Line);

– грамматический и лексический разбор (LIPT и StarNT [13], Lexical Attraction [14]).

Другой полезной дополнительной техникой является использование *разреженных контекстов* (sparse contexts), которые представляют собой последовательности сообщений, чередующихся с пустыми позициями в различных комбинациях, например, имеющих вид $s_m \square s_{m-2} \ldots s_6 \square \square s_3 \square s_2 s_1$, где m – длинна контекста; s_i – фиксированные значения сообщений; символом $\square$ отмечены свободные позиции. Разреженный контекст считается наступившим при совпадении всех фиксированных значений сообщений с сообщениями источника в тех же позициях. Статистика по разреженному контексту является суммарной статистикой по обыкновенным контекстам, попадающим под данный шаблон. Разреженные контексты используются в большинстве современных архиваторов, основанных на КМ, и позволяют «перешагивать» через сообщения-исключения, нарушающие текущие контекстные зависимости (например, коды EOL).

Однако древовидно-контекстные Марковские модели, лежащие в основе всех существующих методов стохастического КМ, имеют два существенных ограничения:

– условные вероятности учитывают влияние только непосредственно прилегающих к оцениваемому сообщению контекстов;

– учитывается влияние только прилегающих слева контекстов (или только прилегающих справа контекстов при обработке данных в обратном порядке).

Как уже было отмечено, большинство работ по усовершенствованию КМ касаются того, как пользоваться статистикой по прилегающим односторонним контекстам, и очень мало внимания уделяется тому, какие еще закономерности могут присутствовать и какая еще статистика может быть использована. Типичные данные обладают некоторыми внутренними зависимостями, легко выявляемыми экспериментально, но «невидимыми» для существующих методов КМ. Далее рассматриваются три возможных пути повышения эффективности КМ, каждый из которых представляет совершенно новый подход к использованию контекстной статистики: *ассоциативное КМ*, *использование удаленных контекстов* и *двустороннее КМ*.

Существуют посылки к разработке методов *ассоциативного контекстного моделирования*, способного учитывать статистику по неактивным в данный момент контекстам. Часто активные контексты являются редкими и содержат недостаточно информации, однако по некоторым характеристикам схожи с определенными неактивными контекстами, более полная статистика которых позволит прогнозировать появление ранее не встречавшихся в данном контексте сообщений.

Ассоциативное КМ в дополнение к обычному объединяет значения сообщений в *ассоциативные группы*. Хорошей иллюстрацией качественной ассоциативной группы могут послужить часто встречающиеся гласные буквы русского текста. Так, например, после обработки некоторой части текста может быть замечено, что буквы О, Е, А, И обладают общим свойством появляться в определенных контекстах (например, после согласных). Допустим, активный контекст

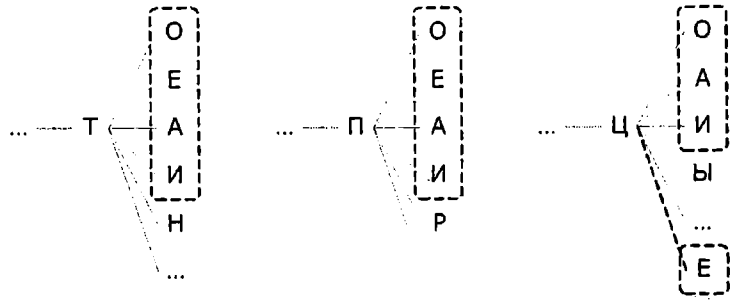

Рис. 1. Пример ассоциативного контекстного моделирования.

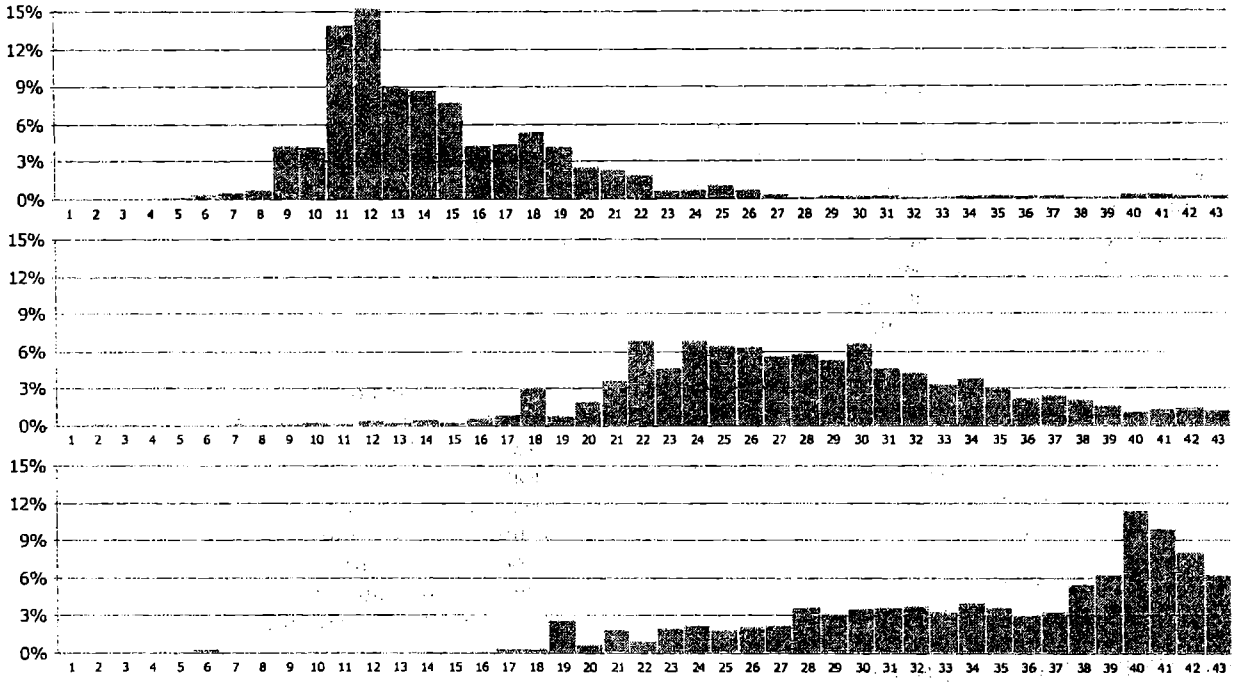

Рис. 2. Гистограммы встречаемости двоеточия, открывающей скобки и закрывающей скобки в 43 позициях после удалённого контекста *__fastcall* в исходных текстах C++.

оканчивается на букву Ц и в данном контексте среди прочих уже встречались буквы О, А, И (рис. 1). Устойчивость ассоциативной группы О-Е-А-И в других контекстах позволяет также ожидать возможное появление буквы Е после буквы Ц.

Значение текущего обрабатываемого сообщения также может быть частично спрогнозировано по серии сообщений, наступивших не непосредственно перед ним, а некоторое время назад — с помощью некоторого *удаленного контекста*. Целесообразность использования такой статистики подтверждается эффективностью разреженных контекстов (см. выше), которые являются частным случаем удаленных контекстов (когда последние позиции являются свободными), либо представляют обобщенную статистику по определенным удаленным контекстам. В общем случае статистика по удаленным контекстам является более дифференцированной, чем по разреженным контекстам, и позволяет делать более точные предсказания.

Полезность использования удаленных контекстов может быть продемонстрирована на следующем примере. На рис. 2 представлены нормированные гистограммы встречаемости двоеточия, открывающей скобки и закрывающей скобки в 43 позициях после модификатора *__fastcall* в 8Мб исходных кодов на языке C++, поставляемых в комплекте Borland C++ Builder 5.5.

Примеры строк из исходных текстов C++, содержащих модификатор *__fastcall*:

```
__fastcall TAppExpert::~TAppExpert(void)
__fastcall TAccessReport::ConnectTo(_ReportPtr
intf)
__fastcall TFormatCondition::GetDunk()
__fastcall TFormMain::TabControlChange(TObject*
Sender)
```

Как следует из примеров, появление модификатора *__fastcall* существенно повышает вероятность скорого появления двоеточий и последующего появления открывающей и закрывающей скобок. Непосредственно прилегающие к первому двоеточию контексты, равно как и к открывающей скобке, встречаются редко и предсказывают данные символы гораздо хуже. Разреженные контексты, упомянутые ранее, также окажутся неэффективными из-за переменной длины, разделяющей *__fastcall* и прогнозируемые символы, а также из-за того, что эта длина достаточно велика.

Кроме этого типичные данные настолько же хорошо, а часто даже лучше, сжимаются в обратном порядке. Данное обстоятельство, а также результаты экспериментов с лексическим притяжением [14], позволяет предполагать целесообразность смешанного *двустороннего контекстного моделирования*, которое использует все активные (как прилегающие, так и удаленные) *левосторонние* (предшествующие текущему сообщению) и *правосторонние* (следующие за сообщением) контексты, динамически меняя направление обработки данных. В упрощенном виде схема

Рис. 3. Пример двустороннего контекстного моделирования.

двустороннего КМ может быть продемонстрирована на примере разбора окончания строки «КОК_КОЛОЛ_КОЛОКОЛ_» (рис. 3).

Пусть символы «КОК_КОЛОЛ_КОЛО» уже обработаны. Следующая буква «К» чрезвычайно плохо предсказывается левосторонними активными контекстами О►, ЛО► и ОЛО► (стрелка указывает направление предсказания), в то время как многими удаленными контекстами может быть хорошо предсказано скорое появление буквы «Л». Если вместо следующей буквы «К» закодировать удаленную букву «Л», то недостающие «КО» смогут быть невероятно эффективно предсказаны с помощью имеющегося теперь правостороннего контекста ◄Л (см. рисунок), т.к. после (по направлению стрелки) правостороннего контекста ◄Л во всех трех случаях следовала «О», а после контекстов ◄О и ◄ОЛ в большинстве случаев встречалась «К». После восстановления «КО» возобновляется обработка оставшихся после «Л» символов с использованием активных левосторонних прилегающих и удаленных контекстов.

Приведенные результаты экспериментальных исследований позволяют сделать вывод о целесообразности разработки новых методов КМ и дальнейших исследований в области сжатия данных вообще. На сегодняшний день неизвестны ни теоретический, ни практический предел сжатия типичных данных и ежегодный прирост степени неискажающего сжатия составляет около одного процента для достаточно большого объема тестовых данных [3]. Достижение большего сжатия возможно только с помощью более качественного моделирования, учитывающего как можно больше закономерностей в исходных данных. Разработка новых эффективных методов моделирования представляет практический интерес не только в системах сжатия информации, но также в криптографии, задачах экстраполяции, прогнозирования, распознавания образов, поиска информации и др. областях.

Литература

1. Rissanen J.J., Langton G.G. Universal Modeling and Coding // IEEE Transactions on Information Theory. – Vol. 27, No.1, 1981. – pp. 12-23.

2. Балашов К. Ю. Сжатие информации: анализ методов и подходов // Препринт / Ин-т техн. кибернетики НАН Беларуси; № 6 – Минск, 2000.

3. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео / Ватолин Д., Ратушняк А., Смирнов М., Юкин В. – М.: ДИАЛОГ-МИФИ, 2002. – 384 с.

4. Cleary J.G., Witten I.H. Data compression using adaptive coding and partial string matching // IEEE Transactions on Communications. – Vol. 32(4), 1984. – pp. 396-402.

5. Howard P.G. The design and analysis of efficient lossless data compression systems: PhD thesis. – Brown Univ., Providence, Rhode Island, 1993.

6. Bloom C. Solving the Problems of Context Modeling – California Institute of Technology, 1998.

7. Шкарин Д. А. Повышение эффективности алгоритма PPM // Проблемы передачи информации. – т. 37, вып. 3, 2001. – с. 44-54.

8. Cleary J.G., Teahan W.J., Witten I.H. Unbounded length contexts for PPM // Proceedings of Data Compression Conference. – IEEE Computer Society, Snowbird Utah, 1995.

9. Потапов В.Н. Арифметическое кодирование вероятностных источников // Дискретная математика и ее приложения: Сборник лекций молодежных научных школ по дискретной математике и ее приложениям. – М.: Изд-во центра прикладных исследований при механико-математическом факультете МГУ, 2001. – 127 с.

10. Потапов В.Н. Обзор методов неискажающего кодирования дискретных источников // Дискретный анализ и исследование операций. – Н.: Изд-во Ин-та матем. им.С.Л.Соболева. – сер.1, т.6, №4, 1999. – с. 49-91.

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

12. Bunton S. On-Line Stochastic Processes in Data Compression: PhD thesis. – Dept. of Comp. Sci., Univ. of Washington, 1997.

13. Sun W., Mukherjee A., Zhang N. A Dictionary-Based Multi-Corpora Text Compression System // Proceedings of Data Compression Conference. – IEEE Computer Society, Snowbird Utah, 2003.

14. Bach J., Witten I.H. Lexical Attraction for Text Compression // Proceedings of Data Compression Conference. – IEEE Computer Society, Snowbird Utah, 1999.

БОГАТОВ Роман Николаевич, аспирант кафедры АСОИУ.