*УДК 550.34.03*

# Методы сжатия данных в геофизических исследованиях

© 2008 г. Д.Б.Калошин, Е.В.Башкирев, В.Ю.Бурмин

*Институт физики Земли им. О.Ю.Шмидта РАН, г. Москва, Россия*

Рассматривается актуальная для геофизических исследований проблема хранения сейсмических данных. С целью уменьшения объема хранимой информации предлагается использовать сжатие данных – удаление избыточности, используя алгоритмы обратимого сжатия (сжатие без потери информации, неискажающее сжатие). Дается обзор современных алгоритмов сжатия и предлагаются рекомендации по использованию алгоритмов при накоплении сейсмических данных, а также при их хранении.

**Ключевые слова:** геофизические исследования, хранение сейсмических данных, сжатие информации.

## Введение

Геофизические и, в частности, сейсмические исследования, ведутся на протяжении многих десятилетий. Легко представить себе ценность накопленной за эти годы информации, и у всякой организации, занимающейся исследованиями в области геофизики, всегда остается насущным вопрос хранения полученных данных.

Так для чего же необходимо длительное сохранение геофизических данных? Наука не стоит на месте и по прошествии времени появляются новые методы и инструменты для обработки полученных ранее наблюдений. Это позволяет по-новому взглянуть на полученные результаты и, возможно, получить дополнительную информацию о строение и структуре геофизической среды.

Сохранность информации в первую очередь зависит от метода хранения. Вплоть до 60-х годов прошлого века сейсмическую информацию записывали на бумажные носители. Такой носитель, конечно же, был недолговечен и большая часть полученных данных со временем могла быть утрачена. Позднее, в 1970-х–1980-х годах, в сейсмологии и сейсморазведке стали применять запись на магнитную ленту. После обработки лента отправлялась в архив, где хранилась на бобинах. Плотность записанной информации была невелика, и требовались большие помещения для хранения данных. В первом и во втором случаях возникала проблема хранения и использования информации на этих носителях.

Лишь в середине 1990-х годов с развитием технологий хранения данных у исследователей появилась возможность записывать и сохранять на десятилетия огромные объемы важной информации. Но даже с учетом современных технологий (CD, DVD, HD-DVD, Blu-Ray) требуются значительные помещения для хранения данных.

Кроме проблемы хранения информации существует также проблема получения информации. В некоторых случаях процесс получения информации в цифровом виде требует все больше и больше ресурсов. Особенно это касается высокочастотных процессов, для которых степень дискретизации данных оказывается высокой. Так, одним из видов сейсмических наблюдений является регистрация землетрясений с помощью автономных сейсмических станций. При таких наблюдениях автономность станций зависит от двух факторов: первый – энергопотребление аппаратуры и второй – емкость накопителей информации. Очевидно, что количество накапливаемой информации при

автономном режиме можно существенно увеличить за счет сжатия данных. Кроме автономных наблюдений, сжатие данных становится насущным и при передаче их по телекоммуникационным каналам, где стоимость канала пропорциональна количеству передаваемой информации.

Таким образом, задача получения, передачи и хранения геофизической информации в наиболее компактном виде достаточно актуальна. С целью уменьшения размеров хранимой информации и используется сжатие данных. От того, насколько компактно будут представлены смоделированные данные, и определяется эффективность алгоритма сжатия. Сжатие сокращает объем пространства, требуемого для хранения файлов в ЭВМ, и количество времени, необходимого для передачи информации по каналу установленной ширины пропускания. Это есть форма кодирования. Другими целями кодирования являются поиск и исправление ошибок, а также шифрование. Процесс поиска и исправления ошибок противоположен сжатию – он увеличивает избыточность данных, когда их не нужно представлять в удобной для восприятия человеком форме. Удаляя из информации избыточность, сжатие способствует шифрованию, что затрудняет поиск шифра доступным для взломщика статистическим методом.

В статье рассмотрено обратимое сжатие (сжатие без потери информации, неискажающее сжатие), где первоначальная информация может быть в точности восстановлена из сжатого состояния. Необратимое сжатие (сжатие с потерей информации, искажающее сжатие) используется для цифровой записи аналоговых сигналов, таких как человеческая речь или рисунки. Обратимое сжатие особенно важно для текстов, записанных на естественных и на искусственных языках, поскольку в этом случае ошибки обычно недопустимы. Хотя первоочередной областью применения рассматриваемых методов есть сжатие текстов, что отражает и наша терминология, однако авторы предлагают использовать рассматриваемые методы при накоплении сейсмических данных, а также при их хранении.

## Основные способы сжатия

Один из самых ранних и хорошо известных методов сжатия – алгоритм Хаффмана [*Huffman*, 1952; *Мастрюков*, 1993; *Потапов*, 1999; *Ватолин и др.,* 2002], который был и остается предметом многих исследований. Однако в конце 1970-х годов благодаря двум важным переломным идеям он был вытеснен. Одна идея заключалась в открытии метода арифметического кодирования [*Pasco*, 1976; *Rissanen*, 1976, 1979; *Rissanen, Langdon*, 1979; *Rubin*, 1979; *Guazzo*, 1980; *Langdon, Rissanen*, 1982; *Langdon*, 1984; *Мастрюков*, 1994а; *Рябко, Фионов*, 1999; *Ватолин и др.,* 2002], имеющего схожую с кодированием Хаффмана функцию, но обладающего несколькими важными свойствами, которые дают возможность достичь значительного превосходства в сжатии. Другим новшеством был метод Зива–Лемпела [*Ziv, Lempel,* 1977, 1978; *Мастрюков*, 1994б; *Ватолин и др.,* 2002], дающий эффективное сжатие и применяющий подход, совершенно отличный от Хаффмановского и арифметического. Оба этих метода со времени первой публикации значительно усовершенствовались, развились и легли в основу практических высокоэффективных алгоритмов.

Существуют два основных способа проведения сжатия: статистический и словарный. Лучшие статистические методы применяют арифметическое кодирование, лучшие словарные – метод Зива–Лемпела. В статистическом сжатии каждому символу присваивается код, основанный на вероятности его появления в тексте. Высоковероятные символы получают короткие коды, и наоборот. В словарном методе группы последовательных символов, или “фраз”, заменяются кодом. Замененная фраза может быть найдена в некотором “словаре”. Только в последнее время было показано, что любая прак-

тическая схема словарного сжатия может быть сведена к соответствующей статистической схеме сжатия, и найден общий алгоритм преобразования словарного метода в статистический. Поэтому при поиске лучшего сжатия статистическое кодирование обещает быть наиболее плодотворным, хотя словарные методы и привлекательны своей быстротой.

## Моделирование и энтропия

Одним из наиболее важных достижений в теории сжатия за последнее десятилетие явилась идея, впервые высказанная в работе Дж.Риссанена и Г.Ландона "Универсальное моделирование и кодирование" [*Rissanen, Langdon*, 1981], о разделении процесса на две части: на кодировщик, непосредственно производящий сжатый поток битов (кодирование), и на моделировщик, поставляющий ему информацию (моделирование). Моделирование присваивает некоторое значение вероятности символам, а кодирование переводит эти значения вероятности в последовательность битов. К сожалению, последние два понятия нетрудно спутать, поскольку "кодирование" часто используют в широком смысле для обозначения всего процесса сжатия (включая моделирование). Существует разница между кодированием в широком смысле (весь процесс) и в узком (производство потока битов на основании данных модели).

Связь между вероятностями и кодами установлена в теореме Шеннона кодирования источника [*Shannon*, 1948], которая показывает, что символ, ожидаемая вероятность появления которого есть $p$, лучше всего представить $\log_2(p)$ битами. Поэтому символ с высокой вероятностью кодируется несколькими битами, тогда как маловероятный требует многих битов. Можно получить ожидаемую длину кода посредством усреднения всех возможных символов, даваемого формулой

$$S = -\sum p(i) \log_2 p(i) .$$

Значение $S$ называется энтропией распределения вероятности, так как это мера количества порядка (или беспорядка) в символах.

Задача моделирования – оценка вероятности для каждого символа. Из этих вероятностей может быть вычислена энтропия. Очень важно отметить, что энтропия есть свойство модели.

Наилучшая средняя длина кода достигается моделями, в которых оценки вероятности как можно более точны. Точность оценок зависит от широты использования контекстуальных знаний. Например, вероятность нахождения буквы «О» в случайно выбранных образцах русских текстов составляет 8.9%. Это сводится к коду для «О», длиной 3.49 бита. Для контраста, если имеем фразу «ОЛОВО», то оценка вероятности появления буквы «О» будет составлять 60% и ее можно закодировать в 0.736 бита. Большего можно достичь, формируя более точные модели текста.

Модель по существу есть набор вероятностей распределения, по одному на каждый контекст, на основании которого символ может быть закодирован. Контексты называются классами условий, так как они определяют оценки вероятности. Нетривиальная модель может содержать тысячи классов условий.

## Адаптированные и неадаптированные модели

В порядке функционального соответствия декодировщик должен иметь доступ к той же модели, что и кодировщик. Для достижения этого есть три способа моделирования: статичное, полуадаптированное и адаптированное (динамическое).

Статичное моделирование использует для всех текстов одну и ту же модель. Она задается при запуске кодировщика, возможно, на основе образцов типа ожидаемого

текста. Такая же копия модели хранится вместе с декодировщиком. Недостаток состоит в том, что схема будет давать непредсказуемо плохое сжатие всякий раз, когда кодируемый текст не вписывается в выбранную модель, поэтому статичное моделирование используют только тогда, когда важны в первую очередь скорость и простота реализации.

Полуадаптированное моделирование решает эту проблему, используя для каждого текста свою модель, которая строится еще до самого сжатия на основе результатов предварительного просмотра текста (или его образца). Перед тем как окончено формирование сжатого текста, модель должна быть передана раскодировщику. Несмотря на дополнительные затраты по передаче модели, эта стратегия в общем случае оказывается более оптимальной благодаря лучшему соответствию модели тексту.

Адаптированное (или динамическое) моделирование уходит от связанных с этой передачей затратах. Первоначально и кодировщик, и раскодировщик присваивают себе некоторую пустую модель, как если бы все символы были равновероятными. Кодировщик использует эту модель для сжатия очередного символа, а раскодировщик – для его разворачивания, затем оба изменяют свои модели одинаковым образом (например, наращивая вероятность рассматриваемого символа). Следующий символ кодируется и достается на основе новой модели, а затем снова изменяет модель. Кодирование продолжается аналогичным раскодированию образом, которое поддерживает идентичную модель за счет применения такого же алгоритма ее изменения, обеспеченным отсутствием ошибок во время кодирования. Используемая модель, которую к тому же не нужно передавать явно, будет хорошо соответствовать сжатому тексту.

Адаптированные модели представляют собой элегантную и эффективную технику и обеспечивают сжатие, по крайней мере, не худшее, чем производимое неадаптированными схемами. Оно может быть значительно лучше, чем у плохо соответствующих текстам статичных моделей [*Cleary, Witten*, 1984].

## Кодирование

Задача замещения символа с вероятностью $p$ приблизительно $\log_2(p)$ битами называется кодированием. Это узкий смысл понятия, а для обозначения более широкого будем использовать термин “сжатие”. Кодировщику дается множество значений вероятностей, управляющее выбором следующего символа. Он производит поток битов, на основе которого этот символ может быть затем раскодирован, если используется тот же набор вероятностей, что и при кодировании. Вероятности появления любого конкретного символа в различных частях текста может быть разной.

Как уже упоминалось выше, хорошо известный метод кодирования – алгоритм Хаффмана, однако он не годится для адаптированных моделей по двум причинам.

Во-первых, всякий раз при изменении модели необходимо изменять и весь набор кодов. Хотя эффективные алгоритмы делают это за счет небольших дополнительных расходов, им все равно нужно место для размещения дерева кодов. Если его использовать в адаптированном кодировании, то для различных вероятностей распределения и соответствующих множеств кодов будут нужны свои классы условий для предсказывания символа. Поскольку модели могут иметь их тысячи, то хранение всех деревьев кодов становится чрезмерно дорогим. Хорошее приближение к кодированию Хаффмана может быть достигнуто применением разновидности расширяющихся деревьев. При этом представление дерева достаточно компактно, чтобы сделать возможным его применение в моделях, имеющих несколько сотен классов условий.

Во-вторых, алгоритм Хаффмана неприемлем в адаптированном кодировании, поскольку выражает значение $\log_2(p)$ целым числом битов. Это особенно неуместно, ко-

гда один символ имеет высокую вероятность (что желательно и является частым случаем в сложных адаптированных моделях). Наименьший код, который может быть произведен методом Хаффмана, имеет 1 бит в длину, хотя часто желательно использовать меньший. Например, «О» в контексте «ОЛОВО» можно закодировать в 0.736 бита. Код Хаффмана превышает необходимый выход примерно на 36%, делая точное предсказание бесполезным.

Эту проблему можно преодолеть блокированием символов, что делает ошибку при ее распределении по всему блоку соответственно маленькой. Однако это вносит свои проблемы, связанные с расширением алфавита (который теперь есть множество всех возможных блоков).

Концептуально более простой и много более привлекательный подход – арифметическое кодирование. К наиболее важным свойствам арифметического кодирования относятся следующие:

– способность кодирования символа вероятности $p$ количеством битов произвольно близким к $\log_2(p)$;

– на каждом шаге вероятности символов могут быть различными;

– очень незначительный запрос памяти независимо от количества классов условий в модели;

– большая скорость.

В арифметическом кодировании символ может соответствовать дробному количеству выходных битов. В нашем примере, в случае появления буквы «О» он может добавить к нему 0.736 бита. На практике результат должен, конечно, быть целым числом битов, что произойдет, если несколько последовательных высоко вероятных символов кодировать вместе, пока в выходной поток нельзя будет добавить 1 бит. Каждый закодированный символ требует только одного целочисленного умножения и нескольких добавлений. Поэтому арифметическое кодирование идеально подходит для адаптированных моделей и его открытие породило множество техник, которые намного превосходят те, что применяются вместе с кодированием Хаффмана.

Сложность арифметического кодирования состоит в том, что оно работает с накапливаемой вероятностью распределения, требующей внесения для символов некоторой упорядоченности. Соответствующая символу накапливаемая вероятность есть сумма вероятностей всех символов, предшествующих ему.

## Заключение

Применение методов сжатия крайне актуально при сборе и накоплении сейсмических данных в первую очередь на автономных программно-аппаратных комплексах, где объем хранимой информации ограничен размером жесткого диска или карты флэш-памяти. Однако для подобных систем крайне актуальна не только степень сжатия, но и объем вычислительных ресурсов (размер задействованной оперативной памяти, скорость процессора, шины данных и т.п.), задействованных при сжатии потока регистрируемой информации. Для реализации рекомендуется использовать неадаптированные модели или словарные методы сжатия, так как они позволяют получить достаточно хорошую степень сжатия регистрируемых данных, при этом не требовательны к вычислительным ресурсам.

При дальнейшей обработке и хранении информации объем задействованных вычислительных ресурсов не является определяющим при выборе метода сжатия. Как следствие рекомендуется использовать алгоритмы, основанные на арифметическом кодировании, так как они предоставляют более высокие степени сжатия по сравнению с другими алгоритмами.

## Литература

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

*Мастрюков Д.* Сжатие по Хаффману // Монитор. 1993. № 7/8.
http://dsrl.ifz.ru/docs/mastrukov_1993_huffman_pdf.rar.

*Мастрюков Д.* Арифметическое кодирование // Монитор. 1994а. № 1.
http://dsrl.ifz.ru/docs/mastrukov_1994_arith_rtf.rar.

*Мастрюков Д.* Алгоритмы группы LZ // Монитор. 1994б. № 2.
http://dsrl.ifz.ru/docs/mastrukov_1994_lzss_pdf.rar.

*Потапов В.Н.* Обзор методов неискажающего кодирования дискретных источников: Дискретный анализ и исследование операций. Новосибирск: Институт математики им. С.Л.Соболева СО РАН, 1999. Сер. 1. Т. 6, № 4. С.49–91.
http://dsrl.ifz.ru/docs/potapov_1999_obzor_pdf.rar.

*Рябко Б.Я., Фионов А.Н.* Эффективный метод адаптивного арифметического кодирования для источников с большими алфавитами // Проблемы передачи информации. 1999. Т. 35, № 4. С.95–108.
http://dsrl.ifz.ru/docs/ryabko_fionov_1999_large_arith_pdf.rar.

*Cleary J.G., Witten I.H.* A comparison of enumerative and adaptive codes // IEEE Trans. Inf. Theory. 1984. IT-30, N 2. P.306–315.

*Huffman D.A.* A method for the construction of minimum redundancy codes // Proceedings of the Institute of Electrical and Radio Engineers. 1952. V. 40, N 9. P.1098–1101.

*Guazzo M.* A general minimum-redundancy source-coding algorithm // IEEE Trans. Inf. Theory. 1980. IT-26, N 1. P.15–25.

*Langdon G.G.* An introduction to arithmetic coding // IBM J. Res. Dev. 1984. V. 28, N 2. P.135–149.

*Langdon G.G., Rissanen J.J.* A simple general binary source code // IEEE Trans. Inf. Theory. 1982. IT-28, N 9. P.800–803.

*Pasco R.* Source coding algorithms for fast data compression: Ph.D. dissertation / Dept. of Electrical Engineering. Stanford Univ., 1976.

*Rissanen J.J.* Generalized Kraft inequality and arithmetic coding // IBM J. Res. Dev. 1976. V. 20, N 5. P.198–203.

*Rissanen J.J.* Arithmetic codings as number representations // Acta Polytechnic Scandinavia Math. 1979. V. 31, N 12. P.44–51.

*Rissanen J.J., Langdon G.G.* Arithmetic coding // IBM J. Res. Dev. 1979. V. 23, N 2. P.149–162.

*Rissanen J.J., Langdon G.G.* Universal modeling and coding // IEEE Trans. Inf. Theory. 1981. IT-27, N 1. P.12–23.

*Rubin F.* Arithmetic stream coding using fixed precision registers // IEEE Trans. Inf. Theory. 1979. IT-25, N 6. P.672–675.

*Shannon C.E.* A mathematical theory of communication // Bell Syst. Tech. J. 1948. V. 27, N 7. P.398–403.

*Ziv J., Lempel A.* A universal algorithms for sequential data compression // IEEE Trans. Inf. Theory. 1977. IT-23, N 3. P.337–343.

*Ziv J., Lempel A.* Compression of individual sequences via variable-rate coding // IEEE Trans. Inf. Theory. IT-24, 1978. N 5. P.530–536.

*Сведения об авторах*

**КАЛОШИН Денис Борисович** – научный сотрудник ИФЗ РАН, 123995, ГСП-5, Москва, Д-242, ул. Большая Грузинская, д. 10. Тел.: (495)-254-68-95.

**БУРМИН Валерий Юрьевич** – доктор физико-математических наук, заведующий лабораторией ИФЗ РАН, 123995, ГСП-5, Москва, Д-242, ул. Большая Грузинская, д. 10. Тел.: (495)-254-68-95. E-mail: burmin@ifz.ru

# Data Compression Techniques in Geophysical Research

D.B. Kaloshin, E.V. Bashkirev, V.Yu. Burmin

*Schmidt Institute of Physics of the Earth, Russian Academy of Sciences, Moscow, Russia*

**Abstract.** The important to geophysical research problem of seismic data storage is considered. To reduce the volume of stored data it is suggested to get rid of information redundancy using algorithms of reversible compression (compression without information loss). A review of modern compression algorithms is given. Compression data algorithms for seismic data accumulation and storage are recommended.