

Статистические методы сжатия информации

11, ноябрь 2014

Белоус В. В.

УДК: 004.627

Россия, МГТУ им. Н.Э. Баумана

walentina.belous@gmail.com

1. Введение

В настоящее время во многих приложениях активно используются разнообразные методы сжатия информации. Им посвящено большое количество литературы, в частности, [3; 6; 7; 11; 12]. Методы сжатия можно разделить на две категории:

- методы сжатия информации без потерь;
- методы сжатия информации с потерями.

Методы сжатия информации без потерь позволяют полностью восстановить сжатую информацию, в то время как методы сжатия с потерями позволяют достичь значительно большего коэффициента сжатия за счет вносимых искажений. Соответственно методы сжатия с потерями имеют более узкую область применения и используются там, где некоторые искажения допустимы – например, для сжатия мультимедиа-информации.

Методы сжатия с потерями более универсальны. Их можно подразделить на методы статистического сжатия и словарные методы сжатия. Настоящий обзор посвящен методам статистического сжатия. В нем мы рассмотрим метод Шеннона-Фано, метод Хаффмана, метод арифметического кодирования и метод PPM.

2. Метод Шеннона-Фано

Метод Шеннона-Фано (Shannon–Fano coding) является одним из первых методов сжатия информации. Впервые он был предложен независимо друг от друга Клодом Шенноном (в работе [10]) и Робертом Фано. Метод основан на представлении информации с помощью кода переменной длины, в котором символы, встречающиеся с большей частотой, кодируются кодом меньшей длины, а встречающиеся с большей частотой – кодом большей длины, используя таким образом избыточность сообщений, заключающуюся в неравномерном распределении частот символов, для сжатия. Причем такой код префиксный, то есть никакое кодовое слово не является началом (префиксом) какого-либо другого кодового слова. Свойство это влечет однозначную декодируемость такого кода.

Опишем теперь этот метод. Сжимаемый текст будем полагать состоящим из символов некоторого алфавита (в двоичном случае, символ может представлять собой, например, один байт, тогда мощность алфавита равна 256). Сначала вычисляются частоты символов в тексте. Далее производится построение префиксного кода с помощью дерева. Его построение начинается с корня, которому ставится в соответствие весь алфавит. Он разбивается на два подмножества, суммарные частоты символов которых примерно одинаковы. Этим подмножествам ставятся в соответствие две вершины дерева второго яруса, являющиеся детьми корня. Затем каждое из этих подмножеств опять разбивается аналогичным образом и полученным подмножествам ставятся в соответствие вершины третьего яруса. При этом, если подмножество включает единственный элемент, то ему соответствует лист кодового дерева и дальнейшему разбиению такое подмножество не подлежит. Аналогично действуем до тех пор, пока не получим все листья. Построенное дерево следует разметить, т.е. его ребра пометить символами 1 и 0. Коды символов формируются из меток ребер на пути от соответствующего символу листа к корню.

Заметим, что разбиение множества элементов в некоторых случаях может быть произведено различными способами. При этом неудачный выбор разбиения может привести к неоптимальности кода. Из-за этого код Шеннона-Фано не является оптимальным в общем случае, хотя может быть оптимальным в частных случаях.

На сегодняшний день, метод Шеннона-Фано практически не используется и представляет лишь исторический интерес. Подмножеством кодов Шеннона-Фано являются коды Хаффмана.

3. Метод Хаффмана

Метод Хаффмана был впервые предложен в 1952 г. в работе [5]. Этот метод похож на метод Шеннона-Фано.

Как и в предыдущем разделе будем полагать, что текст состоит из символов некоторого алфавита. Сначала вычисляются частоты символов в тексте. Далее исходя из этих частот строится дерево кодирования Хаффмана. При этом используется вспомогательный список свободных вершин. Построение дерева происходит следующим образом.

1. Для каждого символа строится по листу дерева. Каждому листу ставится в соответствие вес равный частоте этого символа. Все листья добавляются в список свободных вершин.
2. Из списка свободных вершин выбираются две вершины с наименьшими весами. Они удаляются из этого списка. Создается вершина дерева, являющаяся отцом двух этих вершин и имеющая вес, равный сумме их весов. Она добавляется в список свободных вершин.
3. Одно ребро инцидентное отцовской вершине помечается двоичной цифрой «0», а другое – двоичной цифрой «1».
4. Повторять шаги начиная с шага 2 до тех пор, пока в списке свободных вершин не останется ровно одна вершина. Она станет корнем дерева.

Кодом символа является последовательность меток ребер на пути от корня к листу, соответствующему этому символу.

Метод Хаффмана находит широкое применение как при сжатии фотографий (в составе стандарта JPEG), видео (в составе стандартов MPEG), так и в архиваторах (например, в PKZIP). Хотя в качестве самостоятельного алгоритма сжатия метод применяется редко, чаще он используется в комплексе с другими алгоритмами сжатия.

4. Арифметическое кодирование

Дальнейшим развитием идеи, лежащей в основе метода Хаффмана, является арифметическое кодирование [8; 9; 12].

Отличие этого алгоритма в том, что он работает с непрерывным отрезком, который мы будем называть рабочим отрезком. Вначале объявим рабочим отрезком единичный отрезок. Расположим на нем точки таким образом, что отношения длин образованных подотрезков к длине всего рабочего отрезка будут равны частотам символов, а каждый такой подотрезок будет соответствовать одному символу.

Теперь возьмем очередной символ сжимаемого текста, выберем соответствующий ему отрезок среди только что сформированных и объявим его рабочим. Разобьем и его вышеприведенным образом и т.д. Отрезок будет постоянно уменьшаться. После того, как мы таким образом обработаем все символы сжимаемого текста, возьмем любое число, принадлежащее рабочему отрезку. Оно и будет закодированным сообщением.

Метод арифметического кодирования позволяет достичь высокого коэффициента сжатия.

5. Метод PPM

Метод PPM (prediction by partial matching) [4] представляет собой метод контекстно-зависимого моделирования ограниченного порядка. Он основан на оценке вероятности символа в зависимости от контекста, т.е. от символов, стоящих перед ним. Если для этой оценки используется контекст длины n , то говорят об использовании контекстно-ограниченной модели порядка n .

В рамках модели порядка n , при $n > 0$, оценка вероятности символа c равна отношению количества раз, которое встретилась конкатенация данного контекста с символом c к количеству раз, которое встретился данный контекст. При использовании модели нулевого порядка, оценка вероятности равна частоте, с которой данный символ встретился в тексте. При использовании модели порядка -1 , вероятность каждого символа оценивается как $1/|A|$, где A – алфавит.

Например, пусть следует закодировать строку 121123123122123 над алфавитом $\{1, 2, 3\}$. Оценим вероятность последнего символа.

В модели порядка 2 вероятность символа «3» оценивается как $2/4$, поскольку контекст длины 2 встретился в строке 4 раза, причем 2 раза в этом контексте имел место сим-

вол «3». Модель первого порядка оценивает вероятность символа «3» как $2/5$. Для модели порядка 0 эта вероятность оценивается как $1/5$, а для модели порядка -1 – как $1/3$.

Рассматриваемый алгоритм работает следующим образом. К алфавиту сжимаемой последовательности добавляется специальный символ – код ухода «ESC». Вводится понятие вероятности ухода, т.е. вероятности, которую имеют еще не появлявшиеся символы. При этом любая модель должна выдавать отличную от нуля оценку вероятности ухода.

Пусть задан максимальный порядок m . Для кодирования очередного символа рассматривается модель порядка m . Если она выдает ненулевую оценку вероятности этого символа, то эта вероятность и используется в качестве исходных данных для его кодирования. Если же модель не может произвести оценку вероятности данного символа, либо эта оценка равна нулю, выдается код ухода и производится следующая попытка оценки вероятности этого символа с помощью модели порядка $m - 1$. Если и эта модель не может произвести такую оценку, то применяется модель порядка $m - 2$ и т.д. Так продолжается до тех пор, пока не будет получена ненулевая оценка вероятности. Получение такой оценки гарантируется моделью порядка -1 . В результате, каждый символ может предваряться несколькими символами ухода. При этом символы кодируются, например, с помощью кода Хаффмана, используя вместо частоты символа, полученную оценку вероятности.

Метод RPM демонстрирует один из самых высоких коэффициентов сжатия, прежде всего, для текстов на естественном языке, что обуславливает весьма широкое его применение. Мы также можем предложить его использовать для сжатия текстов на естественном языке перед шифрованием с помощью различных криптоалгоритмов, например, предложенных в работах [1; 2]. Недостатком метода RPM является медленное декодирование.

6. Заключение

Итак, роизведен обзор основных методов статистического сжатия. К сожалению, за рамками этой статьи остались методы словарного сжатия. Им будет посвящена отдельная статья.

Список литературы

1. Ключарев П.Г. Блочные шифры, основанные на обобщённых клеточных автоматах // Наука и образование. Электронное научно-техническое издание. 2012. № 12.
2. Ключарев П.Г. Клеточные автоматы, основанные на графах Рамануджана, в задачах генерации псевдослучайных последовательностей // Наука и образование. Электронное научно-техническое издание. 2011. № 10. — С. <http://technomag.edu.ru/doc/241308.html>.
3. Сэломон Д. Сжатие данных, изображений и звука. — М. : Техносфера, 2006. — 365 с.
4. Cleary J.G., Witten I. Data compression using adaptive coding and partial string matching // Communications, IEEE Transactions on. 1984. Т. 32. № 4. — С. 396-402.

5. Huffman D.A. A method for the construction of minimum redundancy codes // *proc. IRE.* 1952. T. 40. № 9. — C. 1098-1101.
6. Kimura N., Latifi S. A survey on data compression in wireless sensor networks. : IEEE, 2005. — 8-13.
7. Motta G., Rizzo F., Storer J.A. *Hyperspectral data compression.* : Springer, 2006.
8. Rissanen J. Generalized Kraft inequality and arithmetic coding // *IBM Journal of research and development.* 1976. T. 20. № 3. — C. 198-203.
9. Rissanen J., Langdon Jr G.G. Arithmetic coding // *IBM Journal of research and development.* 1979. T. 23. № 2. — C. 149-162.
10. Shannon K., Weaver W. A mathematical theory of communication // *Bell System Tehn. J.* 1948. T. 3. — C. 623-637.
11. Srisooksai T., Keamarungsi K., Lamsrichan P., Araki K. Practical data compression in wireless sensor networks: A survey // *Journal of Network and Computer Applications.* 2012. T. 35. № 1. — C. 37-59.
12. Witten I.H., Neal R.M., Cleary J.G. Arithmetic coding for data compression // *Communications of the ACM.* 1987. T. 30. № 6. — C. 520-540.