СЖАТИЕ ИНФОРМАЦИИ С ПОМОЩЬЮ АЛГОРИТМОВ АДАПТИВНОГО ВЕРОЯТНОСТНОГО КОДИРОВАНИЯ

Лобанов С.В.

Московский государственный авиационный институт (технический университет)
125871, г. Москва, ГСП, Волоколамское шоссе, д. 4, кафедра № 402 РСУПИ

В настоящее время разработано множество алгоритмов сжатия информации без потерь. Например, только различных модификаций алгоритмов LZ77 и LZ78 насчитывается несколько десятков. Аналогично и для других семейств алгоритмов - хаффмановского и арифметического. Во всех этих семействах алгоритмов, так или иначе, но не всегда в явном виде, используются вероятности появления символов алфавита на текущей позиции последовательности. Следовательно, создание алгоритмов, использующих в явном виде вероятностную структуру информационной последовательности, является важнейшей задачей.

Любой алгоритм сжатия информации можно представить как некоторый преобразователь данных поступающих на его вход. Будем считать, что на входе какого-либо алгоритма сжатия информации имеется последовательность π длиной *N* символов алфавита *X*. Примем, что алфавит *X* состоит из *n* букв ($x_i \in X$, $i = \overline{1,n}$). Рассматриваемый алгоритм сжатия информации преобразовывает эту последовательность символов в некоторую другую, имеющую длину $W \le N$ (если рассматривать ее в том же самом алфавите). Последовательность на выходе алгоритма сжатия называется кодом входной последовательности.

Следует отметить, что не существует алгоритма кодирования информации, который сжимал бы абсолютно любую последовательность данных. Это можно показать простейшими рассуждениями. Допустим, что создан некоторый «суперархиватор», который сжимает любую поданную на его вход последовательность на один символ. Пусть на вход такого «суперархиватора» подаются последовательности длинной *N*. Очевидно, что всего возможно n^N разных входных последовательностей. Так как «суперархиватор» сжимает каждую из входных последовательностей на один символ, т. е. из последовательности длиной *N* символов создает последовательность длиной *N*-1 символов, то в этом случае возможно всего n^{N-1} разных кодовых (выходных) последовательностей. Но тогда очевидно, что некоторые из входных последовательностей, количество которых составляет $n^N - n^{N-1}$, при кодировании неизбежно будут преобразованы в одну и ту же выходную последовательность. Естественно, что при декодировании они правильно восстановлены не будут, т. к. декодер заменит их одной и той же последовательностью. Следовательно, такой «суперархиватор» будет являться алгоритмом сжатия с потерями.

Из рассмотрения работы «суперархиватора» становится ясно, что некоторые из входных последовательностей длины *N*, общее количество которых составляет $n^N - n^{N-1}$, сжиматься не могут в принципе. Каким бы мощным ни был алгоритм кодирования, длины кодовых последовательностей для этой группы входных последовательностей останется прежней равной *N*. Можно предположить, что эта группа входных последовательностей относится к так называемым абсолютно случайным последовательностям (АСП). Нетрудно записать формулу для расчета количества абсолютно случайных последовательностей среди всех последовательностей некоторой длины *i* в алфавите размером *n* символов:

$$\alpha(n,i) = n^i - n^{i-1} \qquad (1)$$

Условимся считать, что $\alpha(n,1) = n$.

В процессе своей работы идеальный архиватор должен для последовательности длины *N*, поступившей на его вход, создать АСП, которая будет кодом поступившей последовательности. Для каждой возможной входной последовательности длины *N* из всей совокупности размером в n^N сопоставлена своя собственная кодовая АСП некоторой длины. При этом возможно *N* различных длин от 1 до *N*, которые соответствуют *N* различным коэффициентам сжатия. Очевидно, что тогда общее количество АСП длиной от 1 до *N* должно быть равно n^N, т. е. должно быть справедливо следующее равенство

$$\sum_{i=1}^{N} \alpha(n,i) = n^N \qquad (2)$$

С учетом формулы (1) можно убедиться, что равенство (2) соблюдается.

Таким образом, любую последовательность можно представить в виде двух частей:

- Длины кода для данной входной последовательности;
- Кода последовательности, являющегося АСП определенной длины.

Следовательно, если создать алгоритм определения длины кода любой последовательности и генерации АСП заданной длины, соответствующей заданной последовательности, то получится новый алгоритм сжатия информации оптимальный по своей природе.

Выходную последовательность, т. е. код последовательности, можно рассматривать двояко. Во-первых, его можно рассматривать как некоторое целое число $K(\pi)$, удовлетворяющее неравенству

$$0 \le K(\pi) \le M(\pi) - 1, \tag{3}$$

где $M(\pi)$ - мощность источника информации, рассчитанная исходя из входной последовательности π. Во-вторых, если разделить все части выражения (3) на величину $M(\pi)$, то код последовательности можно рассматривать как некоторое действительное число $B(\pi)$, удовлетворяющее неравенству

$$0 \le B(\pi) = K(\pi)/M(\pi) < 1 \tag{4}$$

В этом случае код последовательности представляется числом с плавающей точкой меньшим единицы и количество знаков после запятой отождествляет его длину.

По своей сути мощность источника информации, рассчитанная исходя из конкретной входной последовательности, показывает число последовательностей, которые с точки зрения алгоритма сжатия информации идентичны по своей вероятностной структуре данной. Можно показать, что мощность дискретного источника информации необходимо вычислять с помощью следующего выражения

$$M(\pi) = \prod_{j=1}^{N} \frac{1}{P_i(x(j)/x(1)x(2)\ldots x(j-1))} \tag{5}$$

Здесь $P_i(x(j)/x(1)x(2)\ldots x(j-1))$ - вероятность появления на *j*-ой позиции последовательности *i*-го символа алфавита, оцененная исходя из *j*-1 предыдущих символов. Очевидно, что чем лучше сжимает алгоритм, тем ближе эти вероятности к единице. Задача алгоритма кодирования заключается в том, чтобы предсказать появление на текущей позиции последовательности определенного символа алфавита и выставить для этого символа максимальное значение вероятности. Для предсказания используются модели источника информации. В настоящее время для этих целей наиболее часто используют методы семейства PPM [2, 4].

Методы семейства PPM могут давать нулевую оценку вероятности появления символа алфавита. Как видно из (5) в этом случае невозможно оценить мощность источника информации. Для выхода из этой ситуации можно использовать следующие выражения для оценки вероятности появления символов алфавита на позициях последовательности

$$P_i(j) = \frac{P_i^*(j) \cdot m + \theta}{m + n \cdot \theta} \tag{6}$$

или

$$P_i(j) = \frac{P_i^*(j) \cdot j + \theta}{j + n \cdot \theta} \tag{7}$$

Здесь $P_i^*(j)$ - вероятность появления символа алфавита, оцененная в соответствии с каким-либо методом. Заметим, эти выражения удовлетворяют требованию нормировки для суммы вероятностей символов алфавита, т. е.

$$\sum_{i=1}^{n} P_i(j) = \sum_{i=1}^{n} P_i^*(j) = 1 \tag{8}$$

В случае, когда выходная последовательность рассматривается как целое число по схеме в соответствии с неравенством (3), то можно получить следующие два способа для расчета кода последовательности:

$$K(\pi)=\sum_{j=1}^{N}\left\{\left[\prod_{k=1}^{j}\frac{1}{P_{x(k)}(k)}\right]^{x(j)-1}\sum_{i=1}^{}P_i(j)\right\} \tag{9}$$

$$K(\pi)=\sum_{j=1}^{N}\left\{\left[\prod_{k=j}^{N}\frac{1}{P_{x(k)}(k)}\right]^{x(j)-1}\sum_{i=1}^{}P_i(j)\right\} \tag{10}$$

Как было указано выше наряду с (3) к определению кода последовательности возможен и другой принцип. Суть его состоит в том, что код последовательности представляется действительным числом меньшим единицы, как было сделано для неравенства (4). Следовательно, если выражения (9) и (10) разделить на $M(\pi)$, определяемое выражением (5), то получим следующие соотношения для расчета кода последовательностей, а именно:

$$B(\pi)=\sum_{j=1}^{N}\left\{\left[\prod_{k=j+1}^{N}P_{x(k)}(k)\right]^{x(j)-1}\sum_{i=1}^{}P_i(j)\right\} \tag{11}$$

$$B(\pi)=\sum_{j=1}^{N}\left\{\left[\prod_{k=1}^{j-1}P_{x(k)}(k)\right]^{x(j)-1}\sum_{i=1}^{}P_i(j)\right\} \tag{12}$$

Подставив $P_i(j)$ в формулу (12) из (6) и (7) и проведя дополнительные преобразования, получим

$$B(\pi)=\sum_{j=1}^{N}\left\{\left[\prod_{k=0}^{j-1}\frac{P^*_{x(k)}(k)\,m+\theta}{m+n\theta}\right]^{x(j)-1}\sum_{i=1}^{}\left(P_i^*(j)\,m+\theta\right)\right\}, \tag{13}$$

если принять, что $P_{x(0)}(0)=(1-\theta)/m$, и соответственно

$$B(\pi)=\sum_{j=1}^{N}\left\{\left[\prod_{k=0}^{j-1}\frac{P^*_{x(k)}(k)\,k+\theta}{k+1+n\theta}\right]^{x(j)-1}\sum_{i=1}^{}\left(P_i^*(j)\;j+\theta\right)\right\} \tag{14}$$

Выражения (13) и (14) есть обобщенная форма адаптивного вероятностного кодирования. При сжатии информации первоначально, когда текущая позиция последовательности $j<m$, необходимо использовать формулу (14). Далее при превышении параметра m переходят к выражению (13). Значение m необходимо выбирать достаточно большим, чтобы проявились асимптотические свойства алгоритма кодирования. В случае, когда параметр $\theta=1$ выражения (13) и (14) существенно упрощаются. С помощью такой схемы кодирования и при использовании точной моделирующей схемы, например одного из метода PPM, можно добиться максимально возможного коэффициента сжатия информации.

Литература

1. Амелькин В. А. Методы нумерационного кодирования. – Новосибирск: Наука, 1986 г. – 158 с.

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

3. Потапов В. Н. Теория информации. Кодирование дискретных вероятностных источников. Учебное пособие – Новосибирск: Новосиб. гос. унив-тет, 1999 г. – 71 с.

4. Bell T., Witten I. H., Cleary J. G. Modeling for text compression. //ACM Computing Surveys. 1989, Vol. 21, №4, p. 557-591.