# СЖАТИЕ ИНФОРМАЦИИ БЕЗ ПОТЕРЬ С ИСПОЛЬЗОВАНИЕМ ВЕРОЯТНОСТНОГО КОДИРОВАНИЯ

Лобанов С. В.

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

В целях сжатия информации без потерь в настоящее время применяются две разновидности алгоритмов: статистические и эвристические. В статистических алгоритмах, так или иначе, используются оценки вероятностей появления символов алфавита в сжимаемой последовательности. В силу этого они устраняют статистическую избыточность в сжимаемых данных. Например, алгоритм Хаффмана сопоставляет более вероятным символам более короткие префиксные коды, а менее вероятным – более длинные. В алгоритме арифметического сжатия более вероятные символы кодируются меньшим количеством битов и наоборот.

К эвристической группе относится семейство алгоритмов Лемпеля-Зива (LZ). Они основаны на поиске повторяющихся отрезков в сжимаемой последовательности и замене их более короткими кодами, которые являются элементами (адресами) кодовой книги (словаря). Таким образом, эвристические алгоритмы основаны на удалении структурной избыточности в сжимаемых данных. В общем случае, чем чаще участок последовательности появлялся раньше, тем вероятнее, что он появится дальше. Поэтому более часто повторяющиеся участки последовательности получают более короткий код (если используется префиксное кодирование) или один и тот же адрес кодовой книги, привнося тем самым статистическую избыточность в результат, которая затем может быть удалена статистическими методами. Следовательно, для этого семейства алгоритмов справедлив тот факт, что, устраняя структурную избыточность, они не полностью устраняют статистическую.

У каждого из отмеченных методов есть свои недостатки. Недостатком алгоритма Хаффмана является его конечный коэффициент сжатия даже для вырожденных последовательностей и невосприимчивость к структурной избыточности. Более перспективным методом сжатия по сравнению с ним является арифметическое кодирование. Если сравнивать его с алгоритмом Хаффмана, то арифметическое сжатие всегда дает лучший результат при некотором небольшом увеличении временных затрат. Но арифметическое кодирование не использует при сжатии структурную избыточность последовательности вследствие чего оно не всегда эффективно. Семейство алгоритмов LZ имеет несколько недостатков, наиболее существенными из которых являются большой расход памяти и относительно невысокая скорость кодирования-декодирования. Другим существенным недостатком является плохая эффективность сжатия для последовательностей, у которых есть статистическая избыточность, но нет структурной.

Все современные адаптивные алгоритмы сжатия информации функционируют со схемой на рис. 1. У различных методов кодирования блок моделирования источника продуцирует различные объекты или величины: в арифметическом кодировании это количество символов алфавита в последовательности; в алгоритме Хаффмана это префиксные коды для символов алфавита; для алгоритма LZ – состояния словаря. Недостатком всех этих методов является жесткая схема построения блока моделирования, которая не позволяет эффективно функционировать алгоритму во многих практических ситуациях.

Алгоритм адаптивного арифметического кодирования сопоставляет сжимаемой последовательности некоторое число с плавающей точкой ($0 \le K(\pi) < 1$), формируемое в соответствии со следующим выражением

$$K(\pi) = \sum_{t=1}^{N} \left\{ \left( \prod_{k=0}^{t-1} \frac{R_{x(k)}(k)+1}{n+k} \right) \cdot \sum_{i=1}^{I(x(t))-1} \left( R_{X(i)}(t)+1 \right) \right\}, \qquad (1)$$

где $R_i(t)$ - количество *i*-х символов алфавита источника информации в последовательности до *t*-ой позиции; $I(x)$ - индекс, который будет присвоен символу *x*, если расставить все символы алфавита в порядке убывания их количеств в последовательности, т. е. значений $R_i(t)$; $X(i)$ - функция обратная $I(x)$.

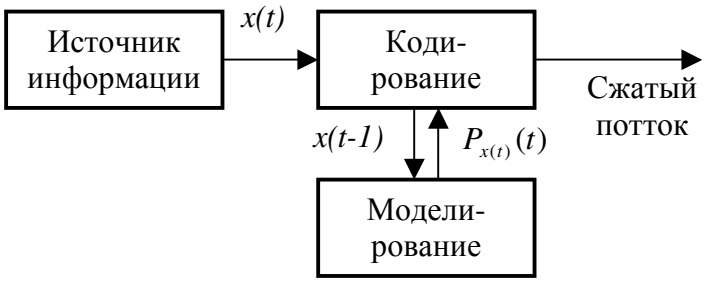


Рис.1 Схема сжатия информации адаптивными алгоритмами.

Недостатком алгоритма арифметического сжатия является жесткая схема кодирования, которая не допускает иные методы моделирования источника информации. Далее предлагается вероятностный подход к сжатию информации, основанный на идеи арифметического кодирования, но свободный от недостатков последнего. Более того, алгоритм арифметического кодирования вытекает из вероятностного алгоритма как частный случай.

Обобщенную формулу адаптивного вероятностного сжатия информации без потерь, где вероятности появления символов алфавита фигурируют в свободном виде, можно представить в следующем виде

$$K(\pi) = \sum_{t=1}^{N} \left\{ \left( \prod_{k=0}^{t-1} \frac{P_{x(k)}(k) \cdot func(k-1)+1}{n+k} \right) \cdot \sum_{i=1}^{I(x(t))-1} \left( P_{X(i)}(t) \cdot (t-1)+1 \right) \right\}. \qquad (2)$$

Здесь $P_i(t)$ - вероятность появления на *t*-ой позиции последовательности *i*-го символа алфавита; $I(x)$ - индекс, который будет присвоен символу *x*, если расставить все символы алфавита источника информации в порядке убывания вероятностей их появления, т. е. значений $P_i(t)$; $X(i)$ - функция обратная $I(x)$; неотрицательная функция

$$func(k) = \begin{cases} 0, & при\ k < 0; \\ k, & при\ k \geq 0. \end{cases}$$

Следует учесть, что, совокупность событий, связанных с появлением на позиции последовательности одного из *n* символов алфавита является полной группой событий, поэтому

$$\sum_{i=1}^{n} P_i(t) = 1.$$

Для реализации алгоритма сжатия на ЭВМ более удобна рекуррентная форма записи. Этот алгоритм допускает такую форму записи, и она выглядит следующим образом

$$K(\pi) = \sum_{t=1}^{N} D(t) \cdot \sum_{i=1}^{I(x(t))-1} \left( P_{X(i)}(t) \cdot (t-1)+1 \right),$$

где $D(t)$ - ядро преобразования, вычисляемое следующим образом

$$D(t+1) = D(t) \cdot \frac{P_{x(t)}(t) \cdot (t-1)+1}{n+t}.$$

Здесь принято, что $D(1) = 1/n$.

Основным при кодировании по обобщенной вероятностной схеме, как следует из выражения (2), является правильность оценки вероятностей появления символов алфавита на текущей позиции последовательности, т. е. построение распределения вероятностей, на основе анализа предыдущих символов, т. е. контекста. Эта задача возложена на модуль моделирования источника, как показано на рис. 1. Основную роль при сжатии информации выполняет именно моделирование источника. В зависимости от эффективности прогнозирования появления символов будет зависеть коэффициент сжатия. Как видно из выражения (2) само кодирование на рис. 1 сводится лишь к вычислениям по этой формуле.

Существуют различные модели источников информации: Бернуллиевская, Пуассоновская, Марковская и др. Наиболее простой моделью источника информации является Бернуллиевская, в которой вероятность появления *i*-го символа алфавита постоянна и не зависит от позиции последовательности, т. е.

$$P_i(t) = P_i = \frac{N_i}{N},$$

где $N_i$ - количество *i*-х символов алфавита в последовательности длиной $N = \sum_{i=1}^{n} N_i$.

Из выражения (2) можно легко получить формулу адаптивного арифметического кодирования (1), если определить вероятность появления *i*-го символа алфавита на *t*-ой позиции последовательности следующим соотношением

$$P_i(t) = \frac{R_i(t)}{t-1}.$$

Такая модель источника информации называется Пуассоновской.

С точки зрения критерия максимального коэффициента сжатия наиболее точной является Марковская модель, в которой учитываются корреляционные связи между символами алфавита. Но одновременно эта модель наиболее сложна для реализации. Кроме того, для нее на сегодняшний день не найдено конкретных формул для построения распределения вероятностей появления символов алфавита. В коммерческих архиваторах наиболее часто применяется контекстуальная модель [1].

Для прогнозирования появления символов на позициях последовательности на практике применяют несколько методик. Наиболее распространенны следующие виды прогнозирования: стохастическое, нейросетевое и с использованием нечеткой логики. Более простое стохастическое прогнозирование основано на экстраполяции контекста на будущее [2]. Этот метод менее трудоемок, но не всегда дает точные результаты при наличии в последовательности скрытых закономерностей. Нейросетевое прогнозирование лишено такого рода недостатков [3]. Но оно требует существенных затрат оперативной памяти и вычислительной мощности ЭВМ для проведения прогнозирования с приемлемой скоростью. Перспективно использование для прогнозирования нечеткой логики, что подтверждается все большим распространением этой теории в различного рода приложениях [4].

Литература.

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

2. Боровиков В. П., Ивченко Г. И. Прогнозирование в системе Statistica в среде Windows. – М.: Финансы и статистика, 1999. – 384 с.

3. Уосерман Ф. Нейрокомпьютерная техника. Теория и практика. – М.: Мир, 1992. – 237 с.

4. Змитрович А. И. Интеллектуальные информационные системы. – Мн.: НТООО «ТетраСистемс», 1997. – 368 с.