Run-Length Encoding

September 8, 2017

1 Algorithm

Run-Length encoding is one of the basic schemes of data compression that do not rely on an explicit probabilistic model of the data. A *run* of letters from a finite alphabet Σ is a maximal contiguous sequence consisting of the same letter. For example, in the string "aaabbaad", there are four runs - "aaaa", "bb", "aa", and "d".

In run-length encoding, we represent each run longer than 2 by the letter followed by that run's length. Runs of length 1 are retained as such.

The compressed version of the above string is "a4b2a2d".

The advantage of this compression for long runs is that $m+1$ occurrences of a letter can be compressed into $\lceil \log m \rceil + \log |A|$ bits, which is significantly shorter.

We introduce certain refinements for convenience in implementation. We assume that the alphabet consists of small English letters encoded in ASCII. We denote this alphabet by A. Each letter can be encoded in a single byte. Instead of identifying runs of arbitrary length, we will encode only runs up to 127 characters long. Longer runs are broken into multiple chunks, each having length at most 127.

This enables easy classification of bytes into letters and lengths. If the leading bit is 0, then it is a letter, else the length of a run.

2 Probabilistic Analysis

In a finite string, the probability of a run of length exactly 2 which starts at position 1 is $\frac{1}{26} \times \frac{25}{26}$, while the probability of a run of length 2 starting at the last position is clearly 0. Thus for finite strings, the expected run length depends on the position i at which we the run starts. For theoeretical convenience, we consider infinite sequences. This will ensure that the average run-length starting at any position is the same.

Lemma 1. *If the letters of the alphabet are distributed uniformly at random, then the average run-length starting at any given position of an infinite sequence in A^∞ is 1.04.*

Proof. The average run length is

$$\sum_{j=1}^{\infty} j \frac{1}{26}^{j-1} \frac{25}{26}, \tag{1}$$

since a run of exactly length j occurs at position i when the letter at position i repeats consecutively for $j-1$ times, and is immediately followed by a different letter to terminate the run. This is an arithmetico-geometric series, with sum 1.04. □

This implies that the run-length encoding *expands* the input strings on average, since the encoding of a run of length 1.04 uses 2 bytes.

If the expected run length is greater than 1, but less than 2, then demonstrably, RLE is wasteful on average.

We now consider a consider probability distributions in which the symbols do not appear uniformly at random. We consider two such examples.

Lemma 2. *Let $p(a) = 4/5$ and the remaining letters have probability $1/125$. Then the expected run length is 6.008.*

Proof. The expected run length is

$$\sum_{j=1}^{\infty} j \left(\frac{4}{5}\right)^{j-1} \frac{1}{5} + j + 1 \left(\frac{1}{125}\right)^{j-1} \frac{124}{125}, \tag{2}$$

where the first term is due to runs of the letter a and the second term due to runs of the other letters, of length j. This value is 6.008. □

In this case, RLE is beneficial on average.

We now look at a different kind of probability distribution. Instead of a letter having a fixed probability regardless of position, we consider a probability distribution where the probability of a letter depends on the previous letter in the sequence.

Lemma 3. *Let the probability of any symbol occurring in the first position of any sequence be 1/26. Subsequently, at any position, suppose the previous letter occurs with probability 4/5 and the remaining letters occur with probability 1/125 each. Then the expected run length is 5.*

Proof. The expected run length starting at any position is

$$\sum_{j=1}^{\infty} j \left(\frac{4}{5}\right) \frac{1}{125}, \tag{3}$$

which is 5. □

The above distribution is an example of what is known as a Markov chain. In this case as well, we can see that RLE is beneficial on average.