Universal and Asymptotically Optimal Compression Algorithms that Operate on the Set of Unbounded Integers

Patrick Peng, Samuel Tian, Alvin Xu

Mentored by Dr. Dan Tamir of Texas State University

ABSTRACT

The field of data compression and encoding has evolved into an ever-growing and ever-important topic, with storage reductions becoming critical in an increasingly data-driven world. In face of these challenges, efforts must be put towards improvements in the techniques used in data compression. In our research, we explore the efficiency of existing encoding schemes for lossless unbounded integer compression, and present two new integer encoding schemes, δ-SFE and δ-RNS, that improve compression efficiency by combining the mathematical principles that existing algorithms use to generate prefix codes. We demonstrate the conformity of our algorithms to several benchmark evaluations of universality and asymptotic optimality, and show the potential advantages these coding schemes offer under certain circumstances based on our experimentation using datasets generated based on various probability mass functions. We further analyze the potential applications of these encoding schemes to key areas such as encryption due to the interchangeability of encoding order in δ-SFE and the usage of prime numbers in δ-RNS.

DEFINITIONS AND THEORY

Definition 1: An **encryption scheme** is a map *P = (G, ρ)*, where *G* is a finite set of characters that consists of the alphabet and M is a finite set of characters that consists of the replacement characters for the alphabet. When applied to information, each element of the plaintext is mapped from its index in *G* to its corresponding element in *ρ*, producing the encrypted text. A probability mass function associated with an encryption scheme is denoted by *p = (G, I)*, where *I* is the set of frequencies of the associated alphabet character in the given plaintext.

Definition 2: A **lossless compression algorithm** is an encryption scheme in which the encrypted text can be unambiguously translated back to the original plaintext. No information is lost in a lossless compression algorithm.

Definition 3: **Entropy**, denoted *H(P)*, is the theoretical minimum average number of bits required to compress the symbols in the data set, given by

$$H(P) = -\sum_{i=0}^{|p|} p_i \log_2 p_i$$

There exists no fixed encryption scheme that can perform better than the entropy.

Definition 4: A **universal code** is an encryption scheme that satisfies the condition that the ratio between the minimal codeword length for the present encoding scheme and the entropy is bounded by a constant. In other words,

$$\frac{E_P(L_\rho)}{\max(1, H(P))} \leq K_\rho$$

where E_p is the expected value of the length L_ρ of one encrypted character.

Definition 5: An **asymptotically optimal code** is a code that satisfies the condition that the ratio R_ρ between the minimal codeword length for the present encoding scheme is a bounded function that approaches 1 as the entropy approaches infinity. In other words,

$$\frac{E_P(L_\rho)}{\max(1, H(P))} \leq R_\rho\big(H(P)\big) \leq K_\rho$$

with

$$\lim_{H \to \infty} R_\rho(H) = 1$$

Definition 6: **Elias-γ** is an encryption scheme that operates on $G = \mathbb{Z}^+$, the set of unbounded integers. If the length of the binary representation of an integer *X* is *N* bits, then we prepend *N-1* zeroes to the binary representation of *X* to yield *γ(X)*. It is known that Elias-γ is a universal encryption scheme, but not asymptotically optimal.

Definition 7: **Elias-δ** is an encryption scheme that operates on $G = \mathbb{Z}^+$. If the length of the binary representation of an integer *X* is *N* bits, then we

OUR ALGORITHMS

δ Shannon Fano Elias Coding

Background: Define a function

$$F(G_x) = \sum_{i<x} p(G_i) + \frac{1}{2} p(G_x)$$

The **Shannon Fano Elias (SFE)** encryption of the alphabet character G_x is then the first $\left\lceil \log_2 \frac{1}{p(G_x)} \right\rceil + 1$ bits to the right of the decimal point in the binary expansion of $F(G_x)$. An additional optimization, known as truncation, can be made to the encryption scheme by erasing the last bit of each code ρ_i until the non-ambiguity condition for lossless compression is invalidated.

δ-SFE: Before any symbols of the plaintext are processed, let *G = {NYT}*, where NYT is a symbol that stands for "not yet transmitted." Let AT be set of "already transmitted" symbols. Note that *G* is a dynamic alphabet, and thus the set *ρ* will also change as more symbols are processed. Our three variations of δ-SFE are defined as follows:

Canonical δ-SFE

$$C\delta{-}SFE(G_i) = \begin{cases} SFE(NYT) + \delta(G_i) & G_i \notin AT \\ SFE(G_i) & G_i \in AT \end{cases}$$

$p(NYT) = \frac{1}{n+1}$, where *n* is the number of symbols (not necessarily unique) that have been processed so far.

Increment δ-SFE

$$I\delta{-}SFE(G_i) = \begin{cases} SFE(NYT) + \delta(G_i) & G_i \notin AT \\ SFE(G_i) & G_i \in AT \end{cases}$$

$p(NYT) = \frac{1+\alpha k}{n+1}$, where α is the number of unique symbols processed and *k* is a predetermined constant. Through experimentation, it seems that *k = 1* is most optimal.

Flagged δ-SFE

$$F\delta{-}SFE(G_i) = \begin{cases} \delta(G_i + 1) & G_i \notin AT \\ 1 + SFE(G_i) & G_i \in AT \end{cases}$$

A binary flag is used to differentiate between occurrences of new and repeated symbols.

δ Residue Number System Coding

Background: A **residue number system (RNS)** is a system where each positive integer x is processed through a set of coprime moduli $\{m_1, m_2, m_3, \ldots, m_n\}$, taking $x \bmod m_k$ for each $k \in (1, n)$. The resulting set $\{r_1, r_2, r_3, \ldots, r_n\}$ is guaranteed to be unique for all

$$x < m_1 m_2 m_3 \ldots m_n.$$

In our tests, we used the set of prime numbers $p = \{2, 3, 5, \ldots\}$ as our set of coprime moduli.

δ-RNS: In δ-RNS, the Elias-δ code is used to compress the numbers generated through RNS. First, y is determined such that y is the least integer such that

$$x < \prod_{b=1}^{y} p_b$$

This number is encoded through Elias-δ coding and forms the start of the string. After determining y, the residues $\{r_1, r_2, r_3, \ldots, r_n\}$ are determined and encoded in binary. Each of these residues r_k is converted to binary and padded with zeroes in front to length $\lceil \log_2 p_k \rceil$ to ensure uniformity. This code is UD and instantaneous. We have also shown it to be universal.

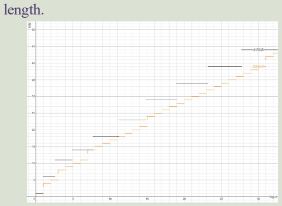
Additionally, due to the nature of a RNS, it provides many advantages in terms of computation, including fast and less resource-intensive operations for creation, addition, multiplication, and subtraction when compared to other encryption methods. These operations can be computed in parallel with a RNS, rather than sequentially, as they can be directly applied to each residue independently. Additionally, sequences of operations are much faster because the modulus of each residue in the system need only be calculated once at the end of a sequence of operations to produce a final set of residues. However, operations such as division and square roots will be more difficult and inefficient.

Due to the fact that a RNS processes each input independently, it produces longer bitstrings because it is not able to use previous information.

METHODOLOGY AND RESULTS

In order to determine the performance of our algorithms, we have generated 4 different plaintext sets, each consisting of 10,000 positive integers following a specific distribution. Data set 1a, 1b, and 1c are all geometric distributions with a probability mass function of $p(G_i) = (1-k)^{G_i - 1} k$, where $k_{1a} = 0.5$, $k_{1b} = 0.1$, and $k_{1c} = 0.01$. Data set 2 satisfies a Poisson distribution with probability mass function $p(G_i) = \frac{\lambda^{G_i} e^{-\lambda}}{G_i!}$, where $\lambda = 128$. Data set 3 consists of pseudo-randomly generated integers from 1 to 1,000. The pseudo-random number generator is derived from Python 2's random library. For each data set, we will compare the performance of Canonical *δ-SFE*, Increment *δ-SFE*, Flagged *δ-SFE*, and *δ-RNS* with entropy and Elias-δ.

To evaluate the performance of δ-RNS further, we compared it with Elias-δ over a range of integers from 0 to approximately 2^{32}. As can be seen from the graph, δ-RNS provides similar performance to Elias-δ in bitstring length.



Our results lead to some intriguing conclusions. Our various δ-SFE resulted in greater performance than Elias-δ in the vast majority of cases, with a greater distinction made in datasets centered towards greater numbers, such as in our Poisson distribution or our pseudo-randomly generated set. This performance approached close to entropy, with greatest compression usually accomplished by Iδ-SFE. These results make these δ-SFE algorithms competitive with conventional compression techniques in terms of practical compression efficiency. These δ-SFE algorithms ultimately also offer unique advantages in their adaptability to dynamic data set distributions as well as the potential application to encryption due to the ability to rearrange the codewords for each symbol into any permutation. δ-RNS performed better than Elias-δ in our Poisson distribution test, and offers the potential for application to encryption due to the explicit usage of prime numbers, a primary component of several encryption schemes. Additionally, δ-RNS has potential in expediting large-scale operations by splitting large integers into sets of smaller integers. Further tests with other datasets will be needed to determine the optimal usage of these various algorithms, as well as to determine the optimal k-value for the Iδ-SFE encoding scheme.

[1] Elias, P. "Universal codeword sets and representations of the integers".IEEE Trans. Inf.Theory21.2 (Mar. 1975): 194–203. Print.

[2] Javed, M. Y. and A. Nadeem. "Data compression through adaptive Huffman coding schemes".2000 TENCON Proceedings(2000). Print.

[3] Katti, R.S. and A. Ghosh. "Security using Shannon-Fano-Elias codes".2009 InternationalSymposium on Circuits and Systems(24-27 May 2009). Print.

[4] Ruan, X. and R. Katti. "Using Improved Shannon-Fano-Elias Codes for Data Encryption".2006 IEEE International Symposium on Information Theory(9-14 July 2006). Print.

[5] Tamir, D. "Delta-Huffman Coding of Unbounded Integers".2018 Data Compression Con-ference(27-30 March 2018). Print.