Universal code (data compression)

From Wikipedia, the free encyclopedia

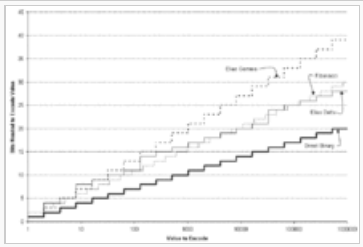
Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

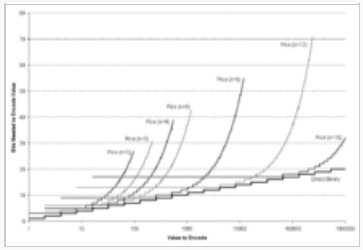
Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Français
한국어
Русский
Edit links



Fibonacci, Elias Gamma, and Elias Delta vs binary coding



Rice with $k = 2, 3, 4, 5, 8, 16$ versus binary

In data compression, a **universal code** for integers is a prefix code that maps the positive integers onto binary codewords, with the additional property that whatever the true probability distribution on integers, as long as the distribution is monotonic (i.e., $p(i) \geq p(i+1)$ for all positive i), the expected lengths of the codewords are within a constant factor of the expected lengths that the optimal code for that probability distribution would have assigned. A universal code is *asymptotically optimal* if the ratio between actual and optimal expected lengths is bounded by a function of the information entropy of the code that, in addition to being bounded, approaches 1 as entropy approaches infinity.

In general, most prefix codes for integers assign longer codewords to larger integers. Such a code can be used to efficiently communicate a message drawn from a set of possible messages, by simply ordering the set of messages by decreasing probability and then sending the index of the intended message. Universal codes are generally not used for precisely known probability distributions, and no universal code is known to be optimal for any distribution used in practice.

A universal code should not be confused with universal source coding, in which the data compression method need not be a fixed prefix code and the ratio between actual and optimal expected lengths must approach one. However, note that an asymptotically optimal universal code can be used on independent identically-distributed sources, by using increasingly large blocks, as a method of universal source coding.

Universal and non-universal codes [edit]

These are some universal codes for integers; an asterisk (*) indicates a code that can be trivially restated in lexicographical order, while a double dagger (‡) indicates a code that is asymptotically optimal:

- Elias gamma coding *
- Elias delta coding * ‡
- Elias omega coding * ‡
- Exp-Golomb coding *, which has Elias gamma coding as a special case. (Used in H.264/MPEG-4 AVC)
- Fibonacci coding
- Levenshtein coding * ‡, the original universal coding technique [1]
- Byte coding, also known as comma coding, where a special bit pattern (with at least two bits) is used to mark the end of the code — for example, if an integer is encoded as a sequence of nibbles representing digits in base 15 instead of the more natural base 16, then the highest nibble value (i.e., a sequence of four ones in binary) can be used to indicate the end of the integer.

These are non-universal ones:

- unary coding, which is used in Elias codes
- Rice coding, which is used in the FLAC audio codec and which has unary coding as a special case
- Golomb coding, which has Rice coding and unary coding as special cases.

Their nonuniversality can be observed by noticing that, if any of these are used to code the Gauss–Kuzmin distribution or the Zeta distribution with parameter s=2, expected codeword length is infinite. For example, using unary coding on the Zeta distribution yields an expected length of

$$E(l) = \frac{6}{\pi^2} \sum_{l=1}^{\infty} \frac{1}{l} = \infty.$$

On the other hand, using the universal Elias gamma coding for the Gauss–Kuzmin distribution results in an expected codeword length (about 3.51 bits) near entropy (about 3.43 bits)[2].

Relationship to practical compression [edit]

Huffman coding and arithmetic coding (when they can be used) give at least as good, and often better compression than any universal code.

However, universal codes are useful when Huffman coding cannot be used — for example, when one does not know the exact probability of each message, but only knows the rankings of their probabilities.

Universal codes are also useful when Huffman codes are inconvenient. For example, when the transmitter but not the receiver knows the probabilities of the messages, Huffman coding requires an overhead of transmitting those probabilities to the receiver. Using a universal code does not have that overhead.

Each universal code, like each other self-delimiting (prefix) binary code, has its own "implied probability distribution" given by $p(i)=2^{-l(i)}$ where $l(i)$ is the length of the ith codeword and $p(i)$ is the corresponding symbol's probability. If the actual message probabilities are $q(i)$ and Kullback–Leibler divergence $D_{KL}(q||p)$ is minimized by the code with $l(i)$, then the optimal Huffman code for that set of messages will be equivalent to that code. Likewise, how close a code is to optimal can be measured by this divergence. Since universal codes are simpler and faster to encode and decode than Huffman codes (which is, in turn, simpler and faster than arithmetic encoding), the universal code would be preferable in cases where $D_{KL}(q||p)$ is sufficiently small. [3]

For any geometric distribution (an exponential distribution on integers), a Golomb code is optimal. With universal codes, the implicit distribution is approximately a power law such as $1/n^2$ (more precisely, a Zipf distribution). For the Fibonacci code, the implicit distribution is approximately $1/n^q$, with

$$q = 1/\log_2(\varphi) \simeq 1.44,$$

where φ is the golden ratio. For the ternary comma code (i.e., encoding in base 3, represented with 2 bits per symbol), the implicit distribution is a power law with $q = 1 + \log_3(4/3) \simeq 1.26$. These distributions thus have near-optimal codes with their respective power laws.

External links [edit]

- Data Compression, by Debra A. Lelewer and Daniel S. Hirschberg (University of California, Irvine)
- *Information Theory, Inference, and Learning Algorithms*, by David MacKay, has a chapter on codes for integers, including an introduction to Elias codes.
- Кодирование целых чисел has mostly English-language papers on universal and other integer codes.

v · t · e	Data compression methods	[hide]
Lossless	Entropy type	Unary · Arithmetic · Golomb · Huffman (Adaptive · Canonical · Modified) · Range · Shannon · Shannon–Fano · Shannon–Fano–Elias · Tunstall · **Universal** (Exp-Golomb · Fibonacci · Gamma · Levenshtein)
	Dictionary type	Byte pair encoding · DEFLATE · Lempel–Ziv (LZ77 / LZ78 (LZ1 / LZ2) · LZJB · LZMA · LZO · LZRW · LZS · LZSS · LZW · LZWL · LZX · LZ4 · Statistical)
	Other types	BWT · CTW · Delta · DMC · MTF · PAQ · PPM · RLE
Audio	Concepts	Bit rate (average (ABR) · constant (CBR) · variable (VBR)) · Companding · Convolution · Dynamic range · Latency · Nyquist–Shannon theorem · Sampling · Sound quality · Speech coding · Sub-band coding
	Codec parts	A-law · μ-law · ACELP · ADPCM · CELP · DPCM · Fourier transform · LPC (LAR · LSP) · MDCT · Psychoacoustic model · WLPC
Image	Concepts	Chroma subsampling · Coding tree unit · Color space · Compression artifact · Image resolution · Macroblock · Pixel · PSNR · Quantization · Standard test image
	Methods	Chain code · DCT · EZW · Fractal · KLT · LP · RLE · SPIHT · Wavelet
Video	Concepts	Bit rate (average (ABR) · constant (CBR) · variable (VBR)) · Display resolution · Frame · Frame rate · Frame types · Interlace · Video characteristics · Video quality
	Codec parts	Lapped transform · DCT · Deblocking filter · Motion compensation
Theory	Entropy · Kolmogorov complexity · Lossy · Quantization · Rate–distortion · Redundancy · Timeline of information theory	
	Compression formats · Compression software (codecs)	

Categories: Data compression | Lossless compression algorithms