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
中文
Edit links

Dynamic Markov compression

From Wikipedia, the free encyclopedia

Dynamic Markov compression (**DMC**) is a lossless data compression algorithm developed by Gordon Cormack and Nigel Horspool.[1] It uses predictive arithmetic coding similar to prediction by partial matching (PPM), except that the input is predicted one bit at a time (rather than one byte at a time). DMC has a good compression ratio and moderate speed, similar to PPM, but requires somewhat more memory and is not widely implemented. Some recent implementations include the experimental compression programs hook by Nania Francesco Antonio, ocamyd by Frank Schwellinger, and as a submodel in paq8l by Matt Mahoney. These are based on the 1993 implementation in C by Gordon Cormack.

Contents [hide]

Algorithm [edit]

DMC predicts and codes one bit at a time. It differs from PPM in that it codes bits rather than bytes, and from context mixing algorithms such as PAQ in that there is only one context per prediction. The predicted bit is then coded using arithmetic coding.

Arithmetic coding [edit]

A bitwise arithmetic coder such as DMC has two components, a predictor and an arithmetic coder. The predictor accepts an n-bit input string $x = x_1x_2...x_n$ and assigns it a probability $p(x)$, expressed as a product of a series of predictions, $p(x_1)p(x_2|x_1)p(x_3|x_1x_2)$ ... $p(x_n|x_1x_2...x_{n-1})$. The arithmetic coder maintains two high precision binary numbers, p_{low} and p_{high}, representing the possible range for the total probability that the model would assign to all strings lexicographically less than x, given the bits of x seen so far. The compressed code for x is p_x, the shortest bit string representing a number between p_{low} and p_{high}. It is always possible to find a number in this range no more than one bit longer than the Shannon limit, $\log_2 1/p(x')$*. One such number can be obtained from* p_{high} *by dropping all of the trailing bits after the first bit that differs from* p_{low}*.*

Compression proceeds as follows. The initial range is set to $p_{low} = 0$, $p_{high} = 1$. For each bit, the predictor estimates $p_0 = p(x_i = 0|x_1x_2...x_{i-1})$ and $p_1 = 1 - p_0$, the probability of a 0 or 1, respectively. The arithmetic coder then divides the current range, (p_{low}, p_{high}) into two parts in proportion to p_0 and p_1. Then the subrange corresponding to the next bit x_i becomes the new range.

For decompression, the predictor makes an identical series of predictions, given the bits decompressed so far. The arithmetic coder makes an identical series of range splits, then selects the range containing p_x and outputs the bit x_i corresponding to that subrange.

In practice, it is not necessary to keep p_{low} and p_{high} in memory to high precision. As the range narrows, the leading bits of both numbers will be the same, and can be output immediately.

DMC model [edit]

The DMC predictor is a table which maps (bitwise) contexts to a pair of counts, n_0 and n_1, representing the number of zeros and ones previously observed in this context. Thus, it predicts that the next bit will be a 0 with probability $p_0 = n_0/n = n_0/(n_0 + n_1)$ and 1 with probability $p_1 = 1 - p_0 = n_1/n$. In addition, each table entry has a pair of pointers to the contexts obtained by appending either a 0 or a 1 to the right of the current context (and possibly dropping bits on the left). Thus, it is never necessary to look up the current context in the table; it is sufficient to maintain a pointer to the current context and follow the links.

In the original DMC implementation, the initial table is the set of all contexts of length 8 to 15 bits that begin on a byte boundary. The initial state is any of the 8 bit contexts. The counts are floating point numbers initialized to a small nonzero constant such as 0.2. The counts are not initialized to zero in order to allow values to be coded even if they have not been seen before in the current context.

Modeling is the same for compression and decompression. For each bit, p_0 and p_1 are computed, bit x_i is coded or decoded, the model is updated by adding 1 to the count corresponding to x_i, and the next context is found by traversing the link corresponding to x_i.

Adding new contexts [edit]

DMC as described above is equivalent to an order-1 context model. However, it is normal to add longer contexts to improve compression. If the current context is A, and the next context B would drop bits on the left, then DMC may add (clone) a new context C from B. C represents the same context as A after appending one bit on the right as with B, but without dropping any bits on the left. The link from A will thus be moved from B to point to C. B and C will both make the same prediction, and both will point to the same pair of next states. The total count, $n = n_0 + n_1$ for C will be equal to the count n_x for A (for input bit x), and that count will be subtracted from B.

For example, suppose that state A represents the context 11111. On input bit 0, it transitions to state B representing context 110, obtained by dropping 3 bits on the left. In context A, there have been 4 zero bits and some number of one bits. In context B, there have been 3 zeros and 7 ones ($n = 10$), which predicts $p_1 = 0.7$.

State	n_0	n_1	next$_0$	next$_1$
A = 11111	4		B	
B = 110	3	7	E	F

C is cloned from B. It represents context 111110. Both B and C predict $p_1 = 0.7$, and both go to the same next states, E and F. The count for C is $n = 4$, equal to n_0 for A. This leaves $n = 6$ for B.

State	n_0	n_1	next$_0$	next$_1$
A = 11111	4		C	
B = 110	1.8	4.2	E	F
C = 111110	1.2	2.8	E	F

States are cloned just prior to transitioning to them. In the original DMC, the condition for cloning a state is when the transition from A to B is at least 2, and the count for B is at least 2 more than that. (When the second threshold is greater than 0, it guarantees that other states will still transition to B after cloning). Some implementations such as hook allow these thresholds to be set as parameters. In paq8l, these thresholds increase as memory is used up to slow the growth rate of new states. In most implementations, when memory is exhausted the model is discarded and reinitialized back to the original bytewise order 1 model.

References [edit]

1. ^ Gordon Cormack and Nigel Horspool, "Data Compression using Dynamic Markov Modelling", Computer Journal 30:6 (December 1987)

External links [edit]

- Data Compression Using Dynamic Markov Modelling
- Google Developers YouTube channel: Compressor Head Episode 3 (Markov Chain Compression) (Page will play audio when loaded)

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: Lossless compression algorithms | Markov models

This page was last modified on 10 June 2014, at 12:26.

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.

Privacy policy About Wikipedia Disclaimers Contact Wikipedia Developers Mobile view