Binary Combinatorial Coding

Vito Dai and Avideh Zakhor
Dept. of Electrical Engineering and Computer Science, U.C. Berkeley
{vdai, avz}@eecs.berkeley.edu

We present a novel binary entropy code called combinatorial coding (CC). The theoretical basis for CC has been described previously under the context of universal coding [1], enumerative coding [2], and minimum description length [3]. The code described in these references works as follows: assume the source data of length M is binary, memoryless, and generated with an unknown parameter θ, the probability that a "1" occurs. To code the data, count the number of "1"s k and encode this using $\lceil \log(M+1) \rceil$ bits. Next, use a ranking algorithm [4] to compute the *index* of the data in a lexicographic listing of all binary sequences of length M with k "1"s. There are $\mathrm{C}(M,k)$ such sequences so the index can be transmitted with $\lceil \log \mathrm{C}(M,k) \rceil$ bits. In general, ranking an M-bit block requires M-bit integer addition and storage. On today's 32-bit computer architectures, $M = 32$ is a practical limitation on block size.

To circumvent this problem, we propose a multi-block CC approach, where the M-bit data sequence is first broken up into N-bit blocks. For each block we compute k and *index* as before, but a simple binary encoding of k is no longer efficient. The distribution of k, which is $Binomial(N,\theta)$, can be approximated with a $Poisson(\lambda)$ distribution where $\lambda = N\theta$. Therefore, we can estimate θ, choose some λ, design a static Huffman code for a $Poission(\lambda)$ distribution, and use it to code k, so as long as we fix $N = \lambda / \theta$. We summarize our design in Table 1, which takes into account factors such as the complexity of ranking and the accuracy of the Poisson approximation.

λ	θ	N	Entropy (bits)	Static Huffman (bits)
8	0.5-0.25	16-32	3.05-3.33	3.12-3.39
4	0.25-0.125	16-32	2.83-2.93	2.89-2.98
2	0.125-0.0625	16-32	2.39-2.43	2.43-2.46
0.8	0.0625-0	13-∞	1.68-1.70	1.80-1.80 (unary code)

Table 1. Properties of the static Huffman code tables used to encode *k* in combinatorial coding

We test the compression efficiency, and encoding and decoding speed of CC against Huffman and arithmetic coding. Tests are performed against a variety of synthetic and "real" data, including DCT coefficients from a H. 263 video codec. The result of one particular test, compressing a binary image of VLSI layout data, is shown in Table 2. Other tests show similar results. Over all the tests, CC achieves the compression efficiency of arithmetic coding, with the coding speed of Huffman coding.

	Uncompressed	Huffman-8	Combinatorial	Arithmetic
Size (kbits)	11370 (1)	1597 (7.2)	230 (49.4)	242 (47.0)
Encode Time (s)	N/A	0.99	0.54	7.46
Decode Time (s)	N/A	0.75	0.56	10.19

Table 2. Result of 3-pixel context based binary image compression, compression ratios are in parenthesis

[1] L. D. Davisson, "Universal Lossless Coding", *IEEE Trans. Of Information Theory*, Nov. 1973.

[2] T. M. Cover, "Enumerative Source Coding", *IEEE Trans. on Information Theory*, Jan. 1973.

[3] M. Hansen, B. Yu, "Model selection and the minimum description length principle", *Journal of American Statistical Association*, vol. 96, June 2001.

[4] A. Nijenhuis, H. S. Wilf, Combinatorial Algorithms, Second Edition, 1978.