Effective Compression of Monotone and Quasi-Monotone Sequences of Integers

Daniel S. Hirschberg and Pierre Baldi
University of California, Irvine
{dan,pfbaldi}@ics.uci.edu

We develop a new class of algorithms for losslessly compressing integer sequences that are monotone or quasi-monotone. We combine aspects of standard entropy codes as expressed in Binary Adaptive Sequential Coding (BASC) [1] and Monotone Length (MOL) coding [2], with an aspect of Binary Interpolative (BI) coding [3].

This development is motivated by the problem of compressing long chemical fingerprint vectors in large databases. The molecular feature frequencies are approximately power-law distributed. When the features are in random (unsorted) order, the gaps between 1-bits tend to an exponential distribution. But if the features are in (sorted) frequency order then the gaps tend to a power-law distribution. Furthermore, in the sorted case, the gaps are quasi-increasing, *i.e.*, they increase most of the time.

Algorithms. Golomb and Elias encode j by catenating a preamble that encodes j's scale, and a mantissa. We can save bits by encoding only the *increases* of the scale. In MOL coding, a 1-bit signals that a gap's scale is at most the previous gap's scale. Otherwise, the number of 0-bits is the scale's increase. Thus, the scale is monotone.

BASC defaults each integer's scale to be the needed scale of the previous integer, allowing the used scale to decrease or increase, whereas MOL disallows decreases. BASC with damping (BASCd) allows the default scale to change by at most one from the previous default. BASC smoothed (BASCs) averages the previously needed scales. A new damping approach, MOLD, allows the default to change up or down, but undoes any change by one.

BI encodes a sequence of increasing integers by encoding the median value using a centered minimal binary code based on knowledge of the median value's possible range, and then recursively encodes the beginning and end subsequence halves.

We combine the MOL/BASC approach with the low-level interpolation aspect of BI coding to create 'I' and '4' hybrids. In MOLI/MOL4 (and BASCI/BASC4, etc.), every second/fourth integer is encoded using MOL (or BASC, etc.) and the intervening skipped integers are then encoded à la BI. If insufficient integers remain to do the necessary skip, encode the last integer and interpolate all skipped integers.

Results. On the ChemDB database, MOL4 was best, enabling us to compress fingerprint vectors, comprising ∼60k binary components, using only 289.49 bits per molecule on average. This is roughly an 8.5% improvement over BI and a 3.8% improvement over the current state-of-the-art variant of BASC. Next best average was BASCsI at +0.80 more than MOL4. Other algorithms performed as follows:

BASC4	MOLDI	BASCs4	MOLD4	MOLI	BASCdI	BASCI	BASCd4	BASCd	BASCs	MOLD	BASC	BI	MOL
+0.97	+1.01	+1.49	+2.09	+2.10	+3.20	+3.45	+8.00	+14.5	+17.5	+18	+25.7	+27	+33.5

[1] Moffat & Anh. Binary codes for locally homogeneous sequences. *Info.Proc.Ltrs.* (2006) 175-180.
[2] Baldi, Benz, Hirschberg, & Swamidass. Lossless compression of chemical fingerprints using integer entropy codes improves storage and retrieval. *J. Chem. Info. and Mod.* (2007) 2098-2109.
[3] Moffat & Stuiver. Binary interpolative coding for effective index compression.*Inf.Retr*(2000)25-47.