Practical Compression with Model-Code Separation

Ying-zong Huang and Gregory W. Wornell [1]

1 Example, Definitions, and Introduction

Suppose an n-bit data source (Source A) takes on two possible values, either one of $\overbrace{0000...00}^{n}$ or $\overbrace{1111...11}^{n}$, with equal probability. A design like Fig. 1 obtains the best compression rate, appears natural and intuitive, yet suffers a surprising degree of systemic inflexibility. What recourse is there (other than total redesign) if the source is only slightly different: say, Markov with a high recurrence probability?

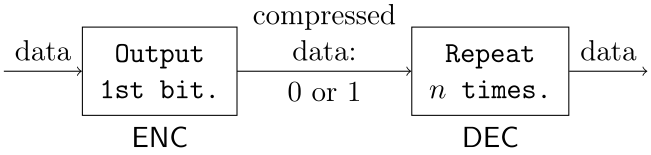

Figure 1: A lossless compression system for Source A.

Inflexibility like this with respect to (any) assumption ultimately stems from a *Joint Model-Code (JMC)* architecture, in which the assignment of compressed output to each data reproduction (*coding*) incorporates a data model (*modeling*) in the process. It does not matter if the data model is learned, adaptive, or mismatched. Nearly all existing systems, lossless and lossy, universal and non-universal, along with the classical random codebook scheme of Shannon source coding theory, are JMC.

2 Separation Architecture with Graphical Message-Passing

We develop an entirely different *Model-Code Separation (MCS)* architecture for fully general compression that has none of the aforementioned inflexibility. In Fig. 2, a *model-free encoder* blindly hashes the data by e.g. random projection to produce interchangeable hashed bits agnostic to any data model or information-preserving data processing, while an *inferential decoder* incorporates a model to recover the original data from among those hashing to the same bits. Such a decoder is practically implementable — we have done so — via low-complexity iterative message-passing algorithms on graphs representing the coding constraint ($\mathcal{C}$) and the data model ($\mathcal{G}$). Performance is not sacrificed vs. JMC.

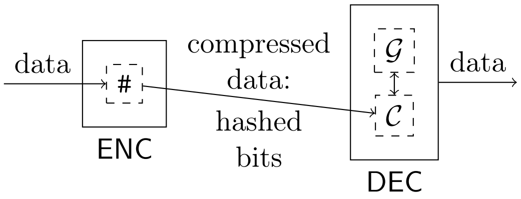

Figure 2: A canonical MCS system with a hashing model-free encoder, a graphical inferential decoder, and interchangeable hashed bits as compressed data.

The advantages of MCS are immediately clear. It retains data model freedom even after compression, and the compressed data on which coding standards are defined can be re-ordered, partially lost, accumulated from multiple origins, etc. These properties support advanced design for emerging complex, dynamic, and machine-learned data types, and for mobile, secure, and network-enabled applications that demand pipeline flexibility. Model-uncertain and lossy compression also have MCS counterparts.

[1] Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, Cambridge, MA 02139. Email: `{zong,gww}@mit.edu`.