Combined Error Correcting and Compressing Codes

Thomas Wenisch, Peter F. Swaszek and Augustus K. Uht[1]
Department of Electrical and Computer Engineering, Kelley Annex
University of Rhode Island, Kingston RI 02881 USA
{iota, swaszek, uht}@ele.uri.edu

I. Introduction

Traditionally, the two encoding operations of compression and error detection/correction are at odds with one another. In the cases where these two encoding operations are combined, they are generally done as two separate encoding steps[1].

Error Correcting and Compression Coding explores the possibility of performing both compression and error correction in a single coding step. ECCC encodes using codewords of varying lengths, but selects codewords such that a minimum Hamming distance between codewords is maintained, enabling error detection and correction.

Four results are presented here. First, a motivating example of ECCC is shown. Second, a generalization of the Kraft-McMillen [1] inequality bounding the existence of ECCC codes is presented. Third, preliminary techniques for generating ECCC codes are explored. Finally, a construction is presented which allows more complex ECCC codes to be built from simple codes.

II. Motivation for ECCC

Though optimal ECCC codes currently cannot be readily generated, ECCC codes have been found which perform well in terms of average codeword length and average density of errors corrected when compared to optimal combinations of Huffman compression and Hamming error correction.

Using a data set from the Calgary Text Compression Corpus [2], the authors discovered an ECCC code which corrects 1 error per encoded source symbol, and encodes source symbols at an average rate of 10.56 bits / symbol. Using a two-step Huffman-Hamming encoding, the data set can be encoded with an error correction rate of 1.57 errors per symbol with a bit rate of 11.01 bits per symbol (7 bit Hamming code packets), or with an error correction rate of 0.57 errors per symbol with a bit rate of 8.58 bits per symbol (11 bit Hamming code packets). As can be seen from these figures, the ECCC code falls between the two options for Huffman-Hamming in terms of the tradeoff between density of errors corrected vs. bit rate.

III. Generalization of Kraft-McMillen

The Kraft-McMillen Inequality establishes a lower bound on the existence of a variable length code given a set of codeword lengths. A similar inequality can be derived for ECCC codes.

Let L represent the length of the longest codeword of a code with q codewords and let S be an L dimensional binary space (with 2^L elements). For the desired error correction, each codeword in an ECCC code must correspond to one of a set of non-overlapping regions in S. Specifically, a codeword of length L occupies the point in S corresponding to that codeword plus all points in S that can be reached by changing up to E_C coordinates (a hypersphere), where E_C is the desired error correction of the ECCC code. Shorter codewords, of length l_i , occupy all points in S whose coordinate representation begins with the codeword or with E_C changes to it. The total number of points in S occupied by such a codeword is:

$$2^{L-l_i}\sum_{j=0}^{E_c}\binom{l_i}{j}$$

For the code to exist, the total number of points occupied by all of the codewords cannot exceed 2^L. Expressed in the form of Kraft-McMillan, this results in the inequality:

$$\sum_{i=0}^{q}\left(2^{-l_i}\sum_{j=0}^{E_c}\binom{l_i}{j}\right)\le 1$$

IV. Preliminary Techniques for Code Generation

The best technique discovered thus far for generating ECCC codes involves maintaining a list of available sequences of a particular length l, and upon selecting a codeword $\boldsymbol{c}$ from that list, deletes all other sequences too close to $\boldsymbol{c}$ for the desired error correction capability. If the set of available sequences seems too small, the list can be grown by increasing l by one bit, doubling the number of available sequences.

The key features of this approach are the criteria used to select a codeword from the list or to increment l. The authors have had some success in generating heuristics for codeword selection, but, thus far, have found no good criterion for choosing when to grow the codeword list.

V. Constructing codes

Valid ECCC codes can be constructed from smaller codes using the following construction:

> Given valid ECCC codes C_1 and C_2 with symbol counts of m and n, respectively, take the last codeword in C_1 and duplicate it n-1 times. Then, to each occurrence of this codeword, append a codeword from C_2. The result will be a valid ECCC code with $m + n - 1$ symbols.

References

[1] R. W. Hamming, *Coding and Information Theory*. Englewood, NJ: Prentice Hall 1980.
[2] T. C. Bell, J. G. Cleary, I. H. Witten, *Text Compression*. Englewood, NJ: Prentice Hall 1990.

[1]This work was partially supported by NSF Grant No. CCR-9708183, and by the URI International Engineering Program.