A Note on the R_0-Parameter for Discrete Memoryless Channels

R. J. McEliece
Communications Systems Research Section

We exhibit an explicit class of discrete memoryless channels (q-ary erasure channels) for which we can design practical and explicit coded systems of rate R with R/R_0 as large as desired.

I. Introduction

For discrete memoryless channels, it is widely agreed that R_0 is second only to channel capacity itself as a quantitative measure of the channel's information-transmission capabilities. It has in fact been conjectured that R_0 represents, in some sense, the "practical limit" to reliable communication on such a channel. In this paper we shall show that this cannot be generally true, by exhibiting an explicit class of discrete memoryless channels (q-ary erasure channels) for which one can design inarguably practical and explicit coded systems of rate R, with R/R_0 as large as desired.

II. The q-ary Erasure Channel

This channel has a q-letter input alphabet X and a q + 1-letter output alphabet Y = XU {?}, where "?" is a special erasure symbol. The transition probabilities are

$$\begin{aligned} p(y|x) &= 1-\epsilon \quad \text{if} \quad x=y \\ &= 0 \quad \text{if} \quad x \neq y \neq ? \\ &= \epsilon \quad \text{if} \quad y = ?, \end{aligned}$$

where $0 \leqslant \epsilon \leqslant 1$ is the erasure probability. Both capacity and R_0 are achieved with an equiprobable input distribution, and an easy calculation gives

$$C = (1-\epsilon)\log q \quad \text{nats per symbol} \qquad (1)$$

$$R_0 = \log\left(\epsilon + (1-\epsilon)q^{-1}\right)^{-1} \quad \text{nats per symbol.} \qquad (2)$$

If q is a power of two, say $q = 2^b$, we can view this channel as a binary channel (with memory) by taking the input letters to be b-bit binary numbers, and by interpreting the erasure symbol "?" as a b-bit erasure burst. Since each transmitted symbol in the original version of this channel corresponds to b transmitted bits in the new version, we have

$$C = (1-\epsilon)\log 2 \quad \text{nats per bit} \qquad (3)$$

$$R_0 = \frac{1}{b}\log\left(\epsilon + (1-\epsilon)2^{-b}\right)^{-1} \quad \text{nats per bit} \qquad (4)$$

We note that, for fixed e,

$$\lim_{b\to\infty} C = \begin{cases} \infty \text{ nats per symbol} \\ (1-\epsilon)\log 2 \text{ nats per bit} \end{cases}$$

$$\lim_{b\to\infty} R_0 = \begin{cases} \log \epsilon^{-1} \text{ nats per symbol} \\ 0 \quad \text{nats per bit} \end{cases}$$

Thus, by taking the alphabet size (burst length) sufficiently large, we can make the ratio C/R_0 as large as desired.

III. Coding for the q-ary Erasure Channel

We consider first the case q = 2, i.e., the binary erasure channel. Linear block codes are especially well suited for this channel (Ref. 1), as we shall now briefly explain. If y is a partially erased codeword from an (n,k) binary block code with parity-check matrix H, to decode y one attempts to express the erased coordinates of y in terms of the unerased coordinates, using elementary row operations on H. The decoding will be successful (i.e., a unique codeword x agreeing with y on its unerased coordinates will be found) if and only if the columns of H corresponding to the erased coordinates of y are linearly independent; but whether successful or not the decoding will require at most r^2 row operations ($r = n - k$ is the code's redundancy) to row-reduce H followed by at most r further row operations to actually compute the values of the erased positions.

Now imagine that we have "built" a "practical" decoder for such a code on a binary erasure channel. We assume the code has rate $k/n = R$, the channel's raw erasure probability is ϵ, and the bit error probability of the decoder is p. We shall now show how to use this code to build an equally practical code for the 2^b-ary erasure channel described in the last section, with the same rate (measured in nats per bit), and with the same decoder bit error probability.

The idea is simply to interleave b copies of the original code. The rate of the interleaved code is the same as the original code, viz., R. The decoding of the interleaved code is actually easier (as measured in computations per decoded bit) than for the original code. This is because the locations of the erasures will be the same for each of the b codewords making up one interleaved block, and so the first step of the decoding, viz., the row-reduction of H, need only be done once. Thus decoding one interleaved block required at most r^2 row operations to row-reduce H followed by at most $b \cdot r$ further row operations to compute the erased components. Since each interleaved block contains bk information bits, the total decoding effort as measured in row operations per decoded bit will be at most A $(r/b + 1)$, where A = $(1 - R)/R$. Thus as the interleaving depth b increases the needed computation per decoded bit slowly decreases to a fixed limit A. We conclude that if the original decoding algorithm was judged to be practical, then the interleaved decoder must also be judged practical. Finally we note that the probability of decoder error is the same for the interleaved and non-interleaved systems, since the interleaved decoder will either decode all b codewords successfully, or none of them.

In summary, given a practical system with R and error probability p for the binary erasure channel, we can construct an equally practical system with the same rate and same error probability for the 2^b-ary erasure channel, for any $b \geq 1$. However, as noted in Section II, the Ro-parameter for these channels approach 0 (nats per bit) as b increases. We thus can design a practical system for which R/R_0 is as large as desired.

For example, take $q = 2^{100}$, ϵ = 0.01. Then C = 99 bits/symbol, R_0 = 6.64 bits/symbol. The $(8,4)$ Hamming code, interleaved to depth 100, will have a decoded bit error probability 6.8 X 10^{-4}, will require at most 1.04 row operations (8.32 bit operations) per decoded bit, and has $R/R_0 = 50/6.64 = 7.5$. If we took $q = 2^{1000}$ instead, we would have $R/R_0 = 75$, everything else being the same.

We conclude that there can be no theorem which relates R/R_0 to decoder complexity.

Reference

1. Berlekamp, E. R., "Error-Correcting Codes," ***Proceedings of the IEEE,*** May 1980, pp. 564-593.