Statistical Approach to ML Decoding of Linear Block Codes on Symmetric Channels

Haris Vikalo and Babak Hassibi
Department of Electrical Engineering
California Institute of Technology
{hvikalo,hassibi}@systems.caltech.edu

Abstract — Maximum-likelihood (ML) decoding of linear block codes on a symmetric channel is studied. Exact ML decoding is known to be computationally difficult. We propose an algorithm that finds the exact solution to the ML decoding problem by performing a depth-first search on a tree. The tree is designed from the code generator matrix and pruned based on the statistics of the channel noise. The complexity of the algorithm is a random variable. We characterize the complexity by means of its first moment, which for binary symmetric channels we find in closed-form. The obtained results indicate that the expected complexity of the algorithm is low over a wide range of system parameters.

I. SUMMARY

We consider transmission over the q-ary symmetric channel. The channel encoder maps the m x 1 information data vector b into the n x 1 codeword c. The encoder employs linear mapping defined via an n x m code generator matrix G, i.e., c = G . b. The receiver observes a corrupted version of the transmitted codeword, r, from which it attempts to recover the information vector b. When the noise is additive, i.e., r = c + v, the ML decoding is equivalent to the nearest codeword problem,

\[ \min_b |r - G \cdot b|, \]  

where |·| denotes Hamming distance. The nearest codeword problem (2) is known to be NP-hard [1].

We propose an algorithm that solves (2) by finding valid codewords within certain Hamming distance d from the observed vector r, i.e., by finding b such that |r - G . b| \leq d. We can choose d according to the statistics of |v|. For brevity, we focus on a binary symmetric channel (BSC). Note that |r - G . b| = |v| = \sum_{i=1}^{n} v_i. Since each v_i is Bernoulli(p), |v| has a binomial distribution and we choose d so that

\[ \sum_{k=0}^{d} \binom{n}{k} p^k (1-p)^{n-k} = 1 - I_p(d+1, n-d) = 1 - \epsilon, \]  

where we set 1 - \epsilon to be close to 1 (so that solution is found with high probability), where \( I_p(a, b) = \frac{B(a, b)}{B(a, b, a)} \) for a \leq b and \( I_p(a, b) = 1 \) otherwise, and where B(a, b) is the beta function, and B(x; a, b) is the incomplete beta function.

Pre-process the code generator matrix G to an approximately upper-triangular form with a diagonal profile as defined by the set of ratios D = \{g_1^{(v)}/g_1^{(h)}, \ldots, g_D^{(v)}/g_D^{(h)}\}, where

\[ g_i^{(v)} = d_i^{(v)} - d_{i-1}^{(v)}, \quad g_i^{(h)} = d_i^{(h)} - d_{i-1}^{(h)}, \quad g_i^{(v)} = d_i^{(v)} - d_{i-1}^{(v)}, \quad g_i^{(h)} = d_i^{(h)} - d_{i-1}^{(h)}. \]

Figure 1: Expected complexity exponent of decoding (R = 1/2, m = 15, n = 30) random binary code.

Figure 1 illustrates expected complexity exponent of the algorithm, defined as \( c_e = \log_m(\text{average flopcount}) \), and compares it with exhaustive search. For small p (say, p < 0.01), the expected complexity of the algorithm is roughly cubic.

REFERENCES


0-7803-8280-3/04/$20.00 ©2004 IEEE 521