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

Haris Vikalo and Babak Hassibi[1]
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 $m \times 1$ information data vector $\mathbf{b}$ into the $n \times 1$ codeword $\mathbf{c}$. The encoder employs linear mapping defined via an $n \times m$ code generator matrix $\mathbf{G}$, i.e., $\mathbf{c} = \mathbf{G} \cdot \mathbf{b}$. The receiver observes a corrupted version of the transmitted codeword, $\mathbf{r}$, from which it attempts to recover the information vector $\mathbf{b}$. When the noise is additive, i.e., $\mathbf{r} = \mathbf{c} + \mathbf{v}$, the *ML decoding* is equivalent to the nearest codeword problem,

$$\min_{\mathbf{b}} |\mathbf{r} - \mathbf{G} \cdot \mathbf{b}|, \tag{1}$$

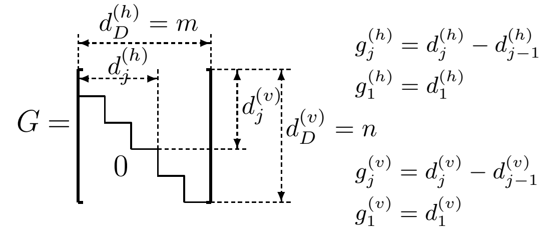
where $|\cdot|$ denotes Hamming distance. The nearest codeword problem (1) is known to be NP-hard [1].

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

$$\sum_{k=0}^{d} \binom{n}{l} p^k (1-p)^{n-k} = 1 - I_p(d+1, n-d) = 1 - \epsilon, \tag{2}$$

where we set $1-\epsilon$ to be close to 1 (so that solution is found with high probability), where $I_x(a,b) = \frac{B(x;a,b)}{B(a,b)}$ for $a \leq b$ and $I_x(a,b) = 1$ 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 $\mathbf{G}$ to an approximately upper-triangular form with a *diagonal profile* as defined by the set of ratios $\mathcal{D} = \{g_1^{(v)}/g_1^{(h)}, \ldots, g_D^{(v)}/g_D^{(h)}\}$, where




Now $|\mathbf{r} - \mathbf{G} \cdot \mathbf{b}| \leq d$ can be written as

$$\sum_{j=1}^{D} \left| \mathbf{r}_j - G_{jj} \cdot \mathbf{b}_j + \sum_{k=j+1}^{D} G_{jk} \cdot \mathbf{b}_k \right| \leq d, \tag{3}$$

where $G_{jk} = G(d_{j-1}^{(v)} + 1 : d_j^{(v)}; d_{k-1}^{(h)} + 1 : d_k^{(h)})$, $\mathbf{b}_j = [b_{d_{j-1}^{(h)}+1} \ \ldots \ b_{d_j^{(h)}}]^T$, and where $\mathbf{r}_j = [r_{d_{j-1}^{(v)}+1} \ \ldots \ \mathbf{r}_{d_j^{(v)}}]^T$, $j = 1, 2, \ldots, D$, $j \leq k \leq D$. We solve (3) with a constrained depth-first tree search similar in spirit to the one in [2]. If no points within distance $d$ is found, $d$ is increased (say, by decreasing $\epsilon$ in (2)) and the algorithm is run anew.

The complexity of the algorithm depends on $G$ and $\mathbf{v}$ and is thus a random variable. Let $f_p(k)$ denote the number of computation per tree node on level $k$. For $G$ with random Bernoulli($\frac{1}{2}$) entries, expected complexity is given by

$$\begin{aligned} C(G,p) \ = \ & \textstyle\sum_{k=1}^{D} f_p(k) \Big[ 1 - I_p(d+1, d_k^{(v)} - d) \\ & + (2^{d_k^{(h)}} - 1) \Big( 1 - I_{\frac{1}{2}}(d+1, d_k^{(v)} - d) \Big) \Big] \end{aligned} \tag{4}$$

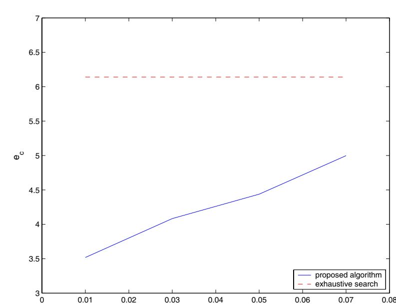



Figure 1: *Expected complexity exponent of decoding* ($R = 1/2, m = 15, n = 30$) *random binary code.*

Figure 1 illustrates expected complexity exponent of the algorithm, defined as $c_e = \log_m$(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

[1] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Transactions on Information Theory*, 24(3):384-386, May 1978.

[2] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of Comput.*, vol. 44, pp. 463-471, April 1985.

[1] This work was supported in part by the National Science Foundation under grant no. CCR-0133818, by the Office of Naval Research under grant no. N00014-02-1-0578, and by Caltech's Lee Center for Advanced Networking.