A dynamical System Approach to Coding Theory: Rational Maps and Maximum Likelihood Decodings

Kazunori Hayashi[*] Yasuaki Hiraoka[†]

Abstract

This article surveys a recent preprint [1], which studies maximum likelihood(ML) decoding in error-correcting codes as rational maps and proposes an approximate ML decoding rule by using a Taylor expansion. The point for the Taylor expansion, which will be denoted by p in the paper, is properly chosen by considering some dynamical system properties. We have two results about this approximate ML decoding. The first result proves that the order of the first nonlinear terms in the Taylor expansion is determined by the minimum distance of its dual code. As the second result, we give numerical results on bit error probabilities for the approximate ML decoding. These numerical results show better performance than that of BCH codes, and indicate that this proposed method approximates the original ML decoding very well.

1 Communication Systems

A mathematical model of communication systems in information theory was developed by Shannon [3]. A general block diagram for visualizing the behavior of such systems is given by Figure 1.1. The source transmits a k-bit message $m=(m_{1}\cdots m_{k})$ to the destination via the channel, which is usually affected by noise e. In order to recover the transmitted message at the destination under the influence of noise, we transform the message into a codeword $x=(x_{1}\cdots x_{n}),\ n\geq k$, by some injective mapping φ at the encoder and input it to the channel. Then the decoder transforms an n-bit received sequence of letters $y=(y_{1}\cdots y_{n})$ by some decoding mapping ψ in order to obtain the transmitted codeword at the destination. Here we consider all arithmetic calculations in some finite field and in this paper we fix it as $\mathbb{F}_{2}=\{0,1\}$. As a model of channels, we deal with a binary symmetric channel (BSC) in this paper which is characterized by the transition probability ϵ $(0<\epsilon<1/2)$. Namely, with probability $1-\epsilon$, the output letter is a faithful replica of the input, and with probability ϵ, it is the opposite of the input letter for each bit (see Figure 1.2). In particular, this is an example of memoryless channels.

Then, one of the main purposes of coding theory is to develop a good encoder-decoder pair (φ, ψ) which is robust to noise perturbations. Hence, the problem is how we efficiently use the redundancy $n\geq k$ in this setting.

[*] Kyoto University: kazunori@i.kyoto-u.ac.jp
[†] Hiroshima University/JST: hiraok@hiroshima-u.ac.jp

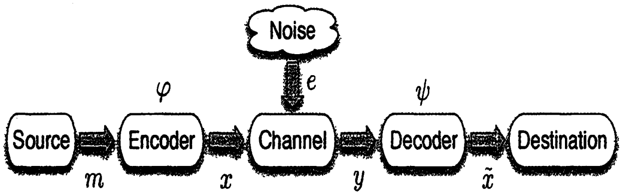

Figure 1.1: Communication system

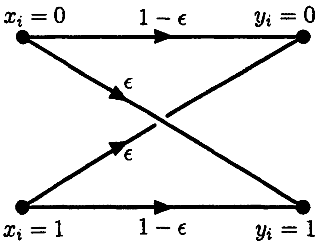

Figure 1.2: Binary symmetric channel

2 Linear Codes

A code with a linear encoding map φ is called a linear code. A codeword in a linear code can be characterized by its generator matrix

$$G = \begin{pmatrix} g_{11} & \cdots & g_{1n} \\ \vdots & \ddots & \vdots \\ g_{k1} & \cdots & g_{kn} \end{pmatrix} = (g_1 \cdots g_n)\,. \quad g_i = \begin{pmatrix} g_{1i} \\ \vdots \\ g_{ki} \end{pmatrix}, \quad i = 1, \cdots, n,$$

where each element $g_{ij} \in \mathbb{F}_2$. Therefore the set of codewords is given by

$$\mathcal{C} = \{(x_1 \cdots x_n) = (m_1 \cdots m_k)G \mid m_i \in \mathbb{F}_2, \}, \quad \sharp\mathcal{C} = 2^k.$$

Here without loss of generality, we assume $g_i \neq 0$ for all $i = 1, \cdots, n$ and rank $G = k$. We call k and n the dimension and the length of the code, respectively.

Because of the linearity, it is also possible to describe $\mathcal{C}$ as a kernel of a matrix H whose $m = n - k$ row vectors are linearly independent and orthogonal to those of G, i.e.,

$$\mathcal{C} = \{(x_1 \cdots x_n) \mid (x_1 \cdots x_n)H^t = 0\}, \quad H = \begin{pmatrix} h_{11} & \cdots & h_{1n} \\ \vdots & \ddots & \vdots \\ h_{m1} & \cdots & h_{mn} \end{pmatrix} = (h_1 \cdots h_n),$$

where H^t means the transpose matrix of H. This matrix H is called a parity check matrix of $\mathcal{C}$. The dual code $\mathcal{C}^*$ of $\mathcal{C}$ is defined in such a way that a parity check matrix of $\mathcal{C}^*$ is given by a generator matrix G of $\mathcal{C}$.

The Hamming distance $d(x, y)$ between two n-bit sequences $x, y \in \mathbb{F}_2^n$ is given by the number of positions at which the two sequences differ. The weight of an element $x \in \mathbb{F}_2^n$ is the Hamming distance to 0, i.e., $d(x) := d(x, 0)$. Then the minimum distance $d(\mathcal{C})$ of a code $\mathcal{C}$ is defined by two different ways as

$$d(\mathcal{C}) = \min\{d(x, y) \mid x, y \in \mathcal{C} \text{ and } x \neq y\} = \min\{d(x) \mid 0 \neq x \in \mathcal{C}\}.$$

Here the second equality results from the linearity. It is easy to observe that the minimum distance is $d(\mathcal{C}) = d$ if and only if there exists a set of d linearly dependent column vectors of H but no set of $d - 1$ linearly dependent column vectors.

For a code $\mathcal{C}$ with the minimum distance $d = d(\mathcal{C})$, let us set $t := \lfloor (d-1)/2 \rfloor$, where $\lfloor a \rfloor$ is the integer part of a. Then, it follows from the following observation that $\mathcal{C}$ can correct t errors: if $y \in \mathbb{F}_2^n$ and $d(x, y) \leq t$ for some $x \in \mathcal{C}$ then x is the only codeword with $d(x, y) \leq t$. In this sense, the minimum distance is one of the important parameters to measure performance of a code and is desirable to design it as large as possible for the robustness to noise.

3 Maximum Likelihood Decoding

Let us recall that, given a transmitted codeword x, the conditional probability $P(y|x)$ of a received sequence $y \in \mathbb{F}_2^n$ at the decoder is given by

$$P(y|x) = P(y_1|x_1) \cdots P(y_n|x_n)$$

for a memoryless channel. Maximum likelihood(ML) decoding $\psi : \mathbb{F}_2^n \ni y \mapsto \tilde{x} \in \mathbb{F}_2^n$ is given by taking the marginalization of $P(y|x)$ for each bit. Precisely speaking, for a received sequence y, the i-th bit element $\tilde{x}_i$ of the decoded word $\tilde{x}$ is determined by the following rule:

$$\tilde{x}_i := \begin{cases} 1, & \sum_{\substack{x \in C \\ x_i=0}} P(y|x) \leq \sum_{\substack{x \in C \\ x_i=1}} P(y|x) \\ 0, & \text{otherwise} \end{cases}, \quad i = 1, \cdots, n. \tag{3.1}$$

In general, for a given decoder ψ, the bit error probability $P_{\text{bit}}^e = \max\{P_1^e, \cdots, P_n^e\}$, where

$$P_i^e = \sum_{x \in C, y \in \mathbb{F}_2^n} P(x,y)(1 - \delta_{x_i, \tilde{x}_i}), \quad \tilde{x} = (\tilde{x}_1, \cdots, \tilde{x}_n) = \psi(y), \quad \delta_{a,b} = \begin{cases} 1, & a = b \\ 0, & a \neq b \end{cases},$$

is one of the important measures of decoding performance. Obviously, it is desirable to design an encoding-decoding pair whose bit error probability is as small as possible. It is known that ML decoding attains the minimum bit error probability for any encodings under the uniform distribution on $P(x)$. In this sense, ML decoding is the best for all decoding rules. However its computational cost requires at least 2^k operations, and it is too much to use for practical applications.

From the above property of ML decoding, one of the key motivation of this work comes from the following simple question. Is it possible to accurately approximate the ML decoding rules with low computational complexity? The main results in this paper give answers to this question.

4 Main Results

Let us first define for each codeword $x \in C$ its codeword polynomial $F_x(u_1, \cdots, u_n)$ as

$$F_x(u_1, \cdots, u_n) := \prod_{i=1}^{n} \rho_i(u_i), \qquad \rho_i(u_i) = \begin{cases} u_i, & x_i = 1, \\ 1 - u_i, & x_i = 0. \end{cases}$$

Then we define a rational map $f : I^n \to I^n$, $I = [0,1]$, by using codeword polynomials as

$$\begin{aligned} & f : (u_1, \cdots, u_n) \mapsto (u_1', \cdots, u_n'), \\ & u_i' = f_i(u) := \frac{\sum_{x \in C, x_i=1} F_x(u)}{H(u)}, \quad i = 1, \cdots, n, \\ & H(u) := \sum_{x \in C} F_x(u), \end{aligned} \tag{4.1}$$

where $u = (u_1, \cdots, u_n)$. This rational map plays the most important role in the paper. It is sometimes denoted by f_G, when we need to emphasize the generator matrix G of the code C.

For a sequence $y \in \mathbb{F}_2^n$, let us take a point $u^0 \in I^n$ as

$$u_i^0 = \begin{cases} \epsilon, & y_i = 0 \\ 1-\epsilon, & y_i = 1 \end{cases}, \quad i = 1, \cdots, n, \tag{4.2}$$

where ϵ is the transition probability of the channel. Then it is straightforward to check that $F_x(u^0) = P(y|x)$. Namely, the conditional probability of y under a codeword $x \in \mathcal{C}$ is given by the value of the corresponding codeword polynomial $F_x(u)$ at $u = u^0$. Therefore, from the construction of the rational map, ML decoding (3.1) is equivalently given by the following rule

$$\begin{aligned} &\psi : \mathbb{F}_2^n \ni y \mapsto \tilde{x} \in \mathbb{F}_2^n, \\ &\tilde{x}_i = \psi_i(y) := \begin{cases} 1, & f_i(u^0) \geq 1/2 \\ 0, & f_i(u^0) < 1/2 \end{cases}, \quad i = 1, \cdots, n. \end{aligned} \tag{4.3}$$

In this sense, the study of ML decoding can be treated by analyzing the image of the initial point (4.2) by the rational map (4.1). Some of the properties of this map in the sense of dynamical systems and coding theory is studied in detail in [1].

For the statement of the main results, we only here mention that this rational map has a fixed point $p := (1/2, \cdots, 1/2)$ for any generator matrix (Proposition 2.2 in [1]). Let us denote the Taylor expansion at p by

$$f(u) = p + Jv + f^{(2)}(v) + \cdots + f^{(l)}(v) + O(v^{l+1}), \tag{4.4}$$

where $v = (v_1, \cdots, v_n)$ is a vector notation of $v_i = u_i - 1/2, i = 1, \cdots, n$, J is the Jacobi matrix at p, $f^{(i)}(v)$ corresponds to the i-th order term, and $O(v^{l+1})$ means the usual order notation. The reason why we choose p as the approximating point is related to the local dynamical property at p and is discussed in [1].

By truncating higher oder terms $O(v^{l+1})$ in (4.4) and denoting it as

$$\tilde{f}(u) = p + Jv + f^{(2)}(v) + \cdots + f^{(l)}(v),$$

we can define the l-th approximation of ML decoding by replacing the map $f(u)$ in (4.3) with $\tilde{f}(u)$, and denote this approximate ML decoding by $\tilde{\psi} : \mathbb{F}_2^n \to \mathbb{F}_2^n$, i.e.,

$$\begin{aligned} &\tilde{\psi} : \mathbb{F}_2^n \ni y \mapsto \tilde{x} \in \mathbb{F}_2^n, \\ &\tilde{x}_i = \tilde{\psi}_i(y) := \begin{cases} 1, & \tilde{f}_i(u^0) \geq 1/2 \\ 0, & \tilde{f}_i(u^0) < 1/2 \end{cases}, \quad i = 1, \cdots, n. \end{aligned} \tag{4.5}$$

Let us remark that the notations $\tilde{f}$ and $\tilde{\psi}$ do not explicitly express the dependence on l for removing unnecessary confusions of subscripts.

4.1 Duality Theorem

We note that there are two different viewpoints on this approximate ML decoding. One way is that, in the sense of its precision, it is preferable to have an expansion with large l. On the other hand, from the viewpoint of low computational complexity, it is desirable to include many zero elements in higher order terms. The next theorem states a sufficient condition to satisfy these two requirements.

Theorem 1 *Let $l \geq 2$. If any distinct l column vectors of a generator matrix G are linearly independent, then the Taylor expansion* (4.4) *at p of the rational map* (4.1) *takes the following form*

$$f(u) = u + f^{(l)}(v) + O(v^{l+1}),$$

where the i-th coordinate $f_i^{(l)}(v)$ of $f^{(l)}(v)$ is given by

$$f_i^{(l)}(v) = \sum_{(i_1,\cdots,i_l)\in\Theta_i^{(l)}} (-2)^{l-1} v_{i_1} \cdots v_{i_l} = -\frac{1}{2} \sum_{(i_1,\cdots,i_l)\in\Theta_i^{(l)}} (1-2u_{i_1})\cdots(1-2u_{i_l}), \tag{4.6}$$

$$\Theta_i^{(l)} = \{(i_1,\cdots,i_l) \mid 1 \leq i_1 < \cdots < i_l \leq n,\ i_k \neq i\ (k=1,\cdots,l),\ g_i + g_{i_1} + \cdots + g_{i_l} = 0\}.$$

First of all, it follows that the larger the minimum distance of the dual code C^* is, the more precise approximation of ML decoding with low computational complexity we have for the code C with the generator matrix G. Especially, we can take $l = d(C^*) - 1$.

Secondly, let us consider the meaning of the approximate map $\tilde{f}(u)$ and its approximate ML decoding $\tilde{\psi}$. We note that each value $u_i^0 (i = 1, \cdots, n)$ in (4.2) for a received word $y \in \mathbb{F}_2^n$ expresses the likelihood $P(y_i|x_i = 1)$. Let us suppose $\Theta_i^{(l)} \neq \emptyset$. Then, from the definition of u^0, each term in the sum of (4.6) satisfies

$$-\frac{1}{2}(1-2u_{i_1})\cdots(1-2u_{i_l}) \begin{cases} <0, & \text{if } y_{i_1} + \cdots + y_{i_l} = 0 \\ >0, & \text{if } y_{i_1} + \cdots + y_{i_l} = 1 \end{cases}, \quad (i_1,\cdots,i_l) \in \Theta_i^{(l)}.$$

When $y_{i_1} + \cdots + y_{i_l} = 0 (= 1, \text{resp.})$, this term decreases(increases, resp.) the value of initial likelihood u_i^0. In view of the decoding rule (4.3), this induces $\tilde{x}_i$ to be decoded into $\tilde{x}_i = 0 (= 1, \text{resp.})$, and this actually corresponds to the structure of the code $g_i + g_{i_1} + \cdots + g_{i_l} = 0$ appearing in $\Theta_i^{(l)}$. In this sense, the approximate map $\tilde{f}(u)$ can be regarded as renewing the likelihood (under suitable normalizations) based on the code structure, and the approximate ML decoding $\tilde{\psi}$ judges these renewed data. From this argument, it is easy to see that a received word $y \in C$ is decoded into $y = \tilde{\psi}(y) \in C$, i.e., the codeword is decoded into itself and, of course, this property should be equipped with any decoders.

We also remark that Theorem 1 can be regarded as a duality theorem in the following sense. Let C be a code whose generator(resp. parity check) matrix is G (resp. H). As we explained in Section 2, the linear independence of the column vectors of H controls the minimum distance $d(C)$ and this is an encoding property. On the other hand, Theorem 1 shows that the linear independence of the column vectors of G, which determines the dual minimum distance $d(C^*)$, controls a decoding property of ML decoding in the sense of accuracy and computational complexity. Hence, we have the correspondence between H/G duality and encoding/decoding duality. This duality viewpoint is discussed in Corollary 4.5 [1] as a setting of geometric Reed-Solomon/Goppa codes.

4.2 Decoding Performance

We show the second result of this paper about the decoding performances of the approximate ML (4.5). For this purpose, let us first examine numerical simulations of the bit error probability on the BSC with the transition probability $\epsilon = 0.16$. We also show numerical results on BCH

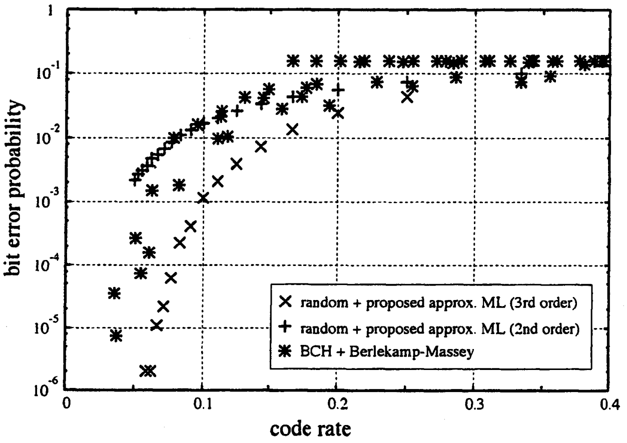

Figure 4.1: BSC with $\epsilon = 0.16$. Horizontal axis: code rate $r = k/n$. Vertical axis: bit error probability. $\times$: random codes with the 3rd order approximate ML decoding. $+$: random codes with the 2nd order approximate ML decoding. $*$: BCH codes with Berlekamp-Massey decoding.

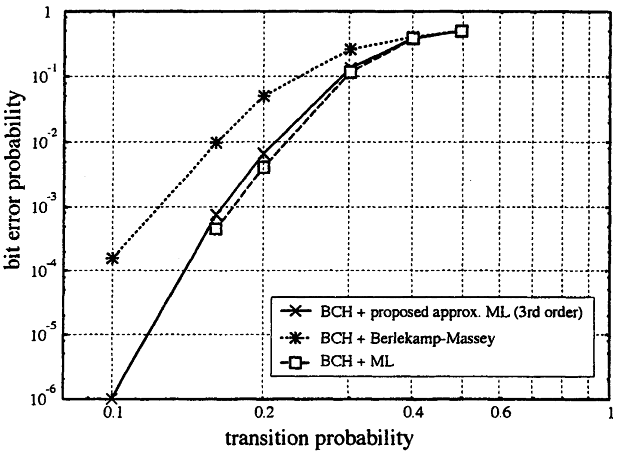

Figure 4.2: Comparison of decoding performances for a BCH code with $k = 7$ and $n = 63$. Horizontal axis: transition probability ϵ. Vertical axis: bit error probability. $\times$: 3rd order approximate ML decoding. $*$: Berlekamp-Massaey decoding. $\Box$: ML decoding.

codes with Berlekamp-Massey decoding for comparison. The results are summarized in Figure 4.1.

Here, the horizontal axis is the code rate $r = k/n$, and the vertical axis is the bit error probability. The plots $+$ ($\times$ resp.) correspond to the 2nd (3rd resp.) order approximate ML (4.5), and $*$ are the results on several BCH codes (n = 7, 15, 31, 63, 127, 255, 511) with Berlekamp-Messey decodings. For the proposed method, we randomly construct a systematic generator matrix in such a way that each column except for the systematic part has the same weight (i.e. number of non-zero elements) w. To be more specific, the submatrix composed by the first k columns of the generator matrix is set to be an identity matrix in order to make the code systematic, while the rest of the generator matrix is made up of $k \times k$ random matrices generated by random permutations of columns of a circulant matrix, whose first column is given by

$$(\underbrace{1, \cdots 1}_{w}, 0, \cdots 0)^t.$$

The reason for using random codings is that we want to investigate average behaviors of the decoding performance, and, for this purpose, we do not put unnecessary additional structure at encodings. The number of matrices added after the systematic part depends on the code rate, and the plot for each code rate corresponds to the best result obtained out of about 100 realizations of the generator matrix. Also, we have employed $w = 3$ and 2 for the generator matrices of the 3rd and the 2nd order approximate ML, respectively. Moreover, the length of the codewords n are assumed to be up to 512. From Figure 4.1, we can see that the proposed method with the 3rd order approximate ML ($\times$) achieves better performance than that of BCH codes with Berlekamp-Massey($*$). It should be also noticed that the decoding performance is improved a lot from the 2nd order to the 3rd order approximation. This improvement is reasonable because of the meaning of the Taylor expansion.

Next, let us directly compare the decoding performances among ML, approximate ML (3rd order) and Berlekamp-Massey by applying them on the same BCH code ($k = 7$, $n = 63$). The result on the bit error probability with respect to transition probability is shown in Figure 4.2. This figure clearly shows that the performance of the 3rd order approximate ML decoding is far better than that of Berlekamp-Massey decoding (e.g., improvement of double-digit at $\epsilon = 0.1$). Furthermore, it should be noted that the 3rd order approximate ML decoding achieves a very close bit error performance to that of ML decoding. Although we have not mathematically confirmed the computational complexity of the proposed approach, the computational time of the approximate ML (3rd order) is much faster than ML decoding. This fact about low computational complexity of the approximate ML is explained as follows: Non-zero higher order terms in (4.6) appear as a result of linear dependent relations of column vectors of G, however, linear dependences require high codimentional scenario. Hence, most of the higher order terms become zeros. As a result, the computational complexity for the approximate ML, which is determined by the number of nonzero terms in the expansion, becomes small.

In conclusion, these numerical simulations suggest that the 3rd order approximate ML decoding approximates ML decoding very well with low computational complexity. We notice that the encodings examined here are random codings. Hence, we can expect to obtain better bit error performance by introducing certain structure on encodings suitable to this proposed decoding rule, or much more exhaustive search of random codes. One of the possibility will be the combination with Theorem 1. On the other hand, it is also possible to consider suitable encoding rules from the viewpoint of dynamical systems via rational maps (4.1). These important subjects are discussed in Section 5 [1] in detail.

Acknowledgment

The authors express their sincere gratitude to the members of TIN working group for valuable comments and discussions on this paper. The authors also wish to thank Shiro Ikeda for pointing out a relationship of this work to information geometry discussed in [2]. This work is supported by JST PRESTO program.

References

[1] K. Hayashi and Y. Hiraoka, Rational Maps and Maximum Likelihood Decodings, http://arxiv.org/pdf/1006.5688v1

[2] S. Ikeda, T. Tanaka, and S. Amari, Information Geometry of Turbo and Low-Density Parity-Check Codes, IEEE Trans. Inform. Theory, vol. 50, pp. 1097-1114, 2004.

[3] C. E. Shannon, A Mathematical Theory of Communication, Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656, 1948.