6.441 Transmission of Information May 16, 2006

Lecture 23

Lecturer: Madhu Sudan *Scribe: Chung Chan*

1 Coding theory

1.1 Introduction

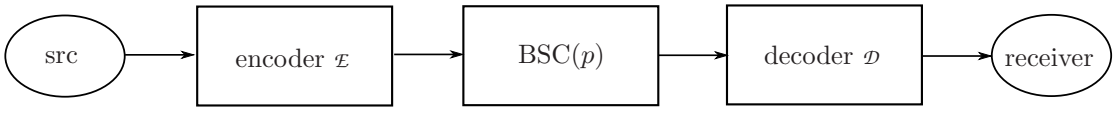


Figure 1: Notations

Consider the binary symmetric channel BSC(p) in Figure 1. Shannon's random coding scheme for any fixed rate $R = 1 - H(p) - \epsilon$ with small ϵ and dimension $k = \lfloor Rn \rfloor$ achieves asymptotically zero error probability $\Pr(\mathcal{E})$ that decays exponentially with the optimal error exponent $E_r(R) = E_{sp}(R) = D[\delta_{GV}(R) \parallel p] = o(\epsilon)$, where $E_r(R), E_{sp}(R)$ and $\delta_{GV}(R)$ are the random ocding exponent, sphere packing error exponent, and Gilbert-Varshamov distance respectively.[1] However, the theorem suggests neither an efficient way of finding the best codebook (other than exhaustive search) nor any special structure of the best code for efficient decoding (other than the computationally intensive ML decoding). Table 1 summarizes the complexity of the naive approach,

	storage	complexity
encoding	$2^k n \doteq 2^{Rn}$ (size of the codebook)	$2^{2^k n} \approx 2^{2^k}$ (number of codebooks for an exhaustive search of the best one)
decoding	$2^k n$ 2^n (size of the decoding table)	2^n (computing the Hamming distance from the observed sequence to each of the valid codeword) efficient (with a decoding table that maps every possible observation sequence to their optimal message hypotheses.)

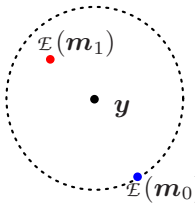
Table 1: Complexity of designing best channel code.

1.2 Smaller ensemble of good codes

In analyzing the probability of error for random code ensemble (RCE), we exploited the pairwise independence property among the randomly chosen codewords $\mathcal{E}(\boldsymbol{m}_0), \mathcal{E}(\boldsymbol{m}_1), \ldots, \mathcal{E}(\boldsymbol{m}_{2^k-1}) \in \{0,1\}^n$ although RCE guarantees a stronger property of mutual independence. Since the stronger property of mutual independence is not needed, we may be able to reduce complexity by relaxing the random code to have only pairwise but not necessarily mutually independent codewords.

How can we have pairwise independence but not mutual independence? Consider the following simpler task of having only onewise independence: choose $\mathcal{E}(M_0), \ldots, \mathcal{E}(M_{2^k-1})$ such that each codeword

[1] To show that the error exponent is $o(\epsilon)$ (as $\epsilon \to 0$), show that $\frac{\partial}{\partial d} D[d \parallel p]\Big|_{d=p} = 0$. For simplicity, we approximate the error exponent by a quadratic function $f(p)\epsilon^2$.



$\Pr[\mathcal{E}(\boldsymbol{m}_1)$ in ball pn around $\boldsymbol{y}|\boldsymbol{y}, \mathcal{E}(\boldsymbol{m}_0)]$ is independent of $\mathcal{E}(\boldsymbol{m}_0)$

Figure 2: How pairwise independence is used in computing $\Pr(\mathcal{E})$

is uniform over the 2^n possibilities. The simplest choice is to set $\mathcal{E}(M_0) = \mathcal{E}(M_1) = \cdots = \mathcal{E}(M_{2^k-1})$ and uniformly distributed.

To have pairwise independence, consider imposing the linearity/affine constraint:

$$\mathcal{E}(\boldsymbol{m}_i) = \boldsymbol{A}^{(k\times n)}\boldsymbol{m}_i^{(k\times 1)} + \boldsymbol{b}^{(n\times 1)}$$

for some randomly chosen $\boldsymbol{A}$ and $\boldsymbol{b}$ where the multiplication and addition are modulo two. If $\boldsymbol{A}$ and $\boldsymbol{b}$ are uniformly random, $\mathcal{E}(\boldsymbol{m}_i)$ and $\mathcal{E}(\boldsymbol{m}_j)$ are independent iff $\boldsymbol{m}_i \neq \boldsymbol{m}_j$. This is true even if $\boldsymbol{b}$ is an all-zero vector because every element of $\mathcal{E}(\boldsymbol{m}_i)$ can be thought of as the corresponding element of $\mathcal{E}(\boldsymbol{m}_j)$ corrupted by a BSC(0.5). If $\boldsymbol{A}$ is a toeplitz matrix, i.e.

$$\boldsymbol{A} = \begin{bmatrix} a_0 & a_{-1} & a_{-2} & \dots & a_{1-n} \\ a_1 & a_0 & a_{-1} & \dots & a_{2-n} \\ a_2 & a_1 & a_0 & \dots & a_{3-n} \\ \vdots & & & & \vdots \\ a_{k-1} & a_{k-2} & a_{k-3} & \dots & a_{k-n} \end{bmatrix}$$

whose entries $a_{1-n}, \dots, a_{k-1}$ are iid Bern(0.5), we also have the desired independence as long as $\boldsymbol{b}$ is uniformly random so that it breaks the dependence among different coordinates.[2]

With the linear random code $\boldsymbol{Am}$ or random affine code $\boldsymbol{Am} + \boldsymbol{b}$ with the Toeplitz matrix $\boldsymbol{A}$, we reduced the numbers of parameters or degrees of freedom to nk and $k+2n-1$ respectively. Effectively, the search space for the best code with the corresponding constraints reduced from doubly exponential 2^{2^k} for RCE to exponential (2^{nk} and 2^{k+2n-1} respectively) without affecting the error exponent *averaged* over the ensemble of codes.[3] The goal now is to further reduce the complexity to polynomial by eliminating parameters of less interest to us.

Consider the following approach,

1. divide the sequence $\boldsymbol{m}$ of k information bits into successive blocks of length $l = 10 \log n$.
2. encode each block separately by the *same* code.

The search space of the best code is polynomial $2^{\frac{l}{R}} = n^{10/R}$ but the probability of error is at least the probability of an error in the first block, which is also at least polynomial $\Pr(\mathcal{E}) \geq 2^{-E_r(R)l/R} = n^{-10E_{sp}(R)/R}$ by the sphere-packing upper bound on error exponent. Is there a way to make the error probability decay exponentially fast in n? The results from Reed Solomon and Peterson in 1960, and Forney in 1966 gives an affirmative answer.

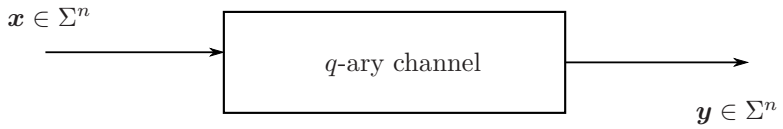
1.3 Concatenation code

Consider the q-ary channel in Figure 3, where Σ is some q-ary alphabet such that with the appropriate definition of addition and multiplication, Σ forms a finite field.[4] This allows us to define polynomials of

[2] To see this, consider the simple $k = n = 2$ case and compare $\mathcal{E}\left(\boldsymbol{m}_0 = \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]\right)$ and $\mathcal{E}\left(\boldsymbol{m}_1 = \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]\right)$.

[3] The best codebooks under the different constraints need not be the same, and hence their error exponent need not be the same.

[4] This requires q to be some positive integral power of a prime number. i.e. $q = (\text{prime})^{(\text{positive integer})}$. In the simple case when q is prime, we can use modulo-q addition and multiplication and the resultant field is called the prime field.



Hamming distance: $\Delta(\boldsymbol{x}, \boldsymbol{y}) := |\{i : 1 \le i \le n, x_i \ne y_i\}|$

Figure 3: q-ary channel

$x \in \Sigma$ in the form of $f(x) := \sum_{i=0}^{k-1} f_i x^i$ (i.e. degree less than k) with $f_i \in \Sigma$ so that they satisfy the fundamental theorem of algebra that a degree $j < k$ polynomial have j roots that are not necessarily distinct. Given $\alpha \in \Sigma$, $f(\alpha) \in \Sigma$ denotes the evaluation of the polynomial f at the point α.

In Reed Solomon code,

1. n distinct points $\beta_1, \ldots, \beta_n \in \Sigma$ are chosen offline and known to both encoder and decoder.
2. The encoder represents the q-ary information k-sequence $f_0, \ldots, f_{k-1}$ as the polynomial $f(x) := \sum_{i=0}^{k-1} f_i x^i$ and then evaluates it at each of the n chosen points. The matrix representation of the evaluation procedure is,

$$\begin{bmatrix} f(\beta_1) \\ f(\beta_2) \\ f(\beta_3) \\ \vdots \\ f(\beta_n) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \beta_1 & \beta_1^2 & \cdots & \beta_1^{k-1} \\ 1 & \beta_2 & \beta_2^2 & \cdots & \beta_2^{k-1} \\ 1 & \beta_3 & \beta_3^2 & \cdots & \beta_3^{k-1} \\ \vdots & & & & \vdots \\ 1 & \beta_n & \beta_n^2 & \cdots & \beta_n^{k-1} \end{bmatrix}}_{\text{Vandermonde matrix}} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_k \end{bmatrix}$$

3. The evaluation sequence $\boldsymbol{x} = [\, f(\beta_1) \cdots f(\beta_n) \,]$ is transmitted through the q-ary channel.
4. The decoder estimates $f_0, \ldots, f_{k-1}$ from $\boldsymbol{y}$ (polynomial interpolation) successfully in polynomial time if $\Delta(\boldsymbol{x}, \boldsymbol{y}) \le \frac{n-k}{2}$.[Peterson 1960]

Roughly speaking, the error-correction capability of the code stems from the structure of polynomial evaluation or the transformation by a Vandermonde matrix. The polynomial decoding time is due to the efficient implementation of finite-field arithmetics. As a sanity check to see how the polynomial structure help recover the information, let us prove that the information sequence $f_0, \ldots, f_{k-1}$ is recoverable if there is no error and $n \ge k$. Suppose, for contradiction, that there exists $f' \ne f$ such that $f'(x) = f(x)$ at the n distinct points. In other words, $f'(x) - f(x)$ is a polynomial with $\deg < k \le n$ but n distinct roots $\beta_1, \ldots, \beta_n$. This contradicts the fundamental theorem of algebra and thus gives the desired result.

Going back to the BSC(p), we can improve the coding by a layered architecture in Figure 4: concatenating an outer 2^l-ary Reed-Solomon code with an inner binary code with the Toeplitz structure as follows

1. divide the k-bit information sequence $\boldsymbol{m}$ into k/l consecutive blocks of length $l := c \log n$ for some constant c.
2. the encoder treat the sequence as one block of k/l 2^l-ary symbol, and uses the Reed-Solomon code to encode it into a block of $(1+\delta)k/l$ 2^l-ary symbols.
3. the encoder now treat the sequence as $(1+\delta)k/l$ blocks of binary l-sequence and uses the same binary code to encode each block to a binary l/R-sequence of length l/R.
4. The entire binary n-sequence $\boldsymbol{x}$ is transmitted through the BSC(p).
5. The decoder receive the binary n-sequence $\boldsymbol{y}$, treat it as k/l blocks of binary l/R-sequence and uses the inner binary code to decode each block to a binary $(1+\delta)k/l$-sequence.

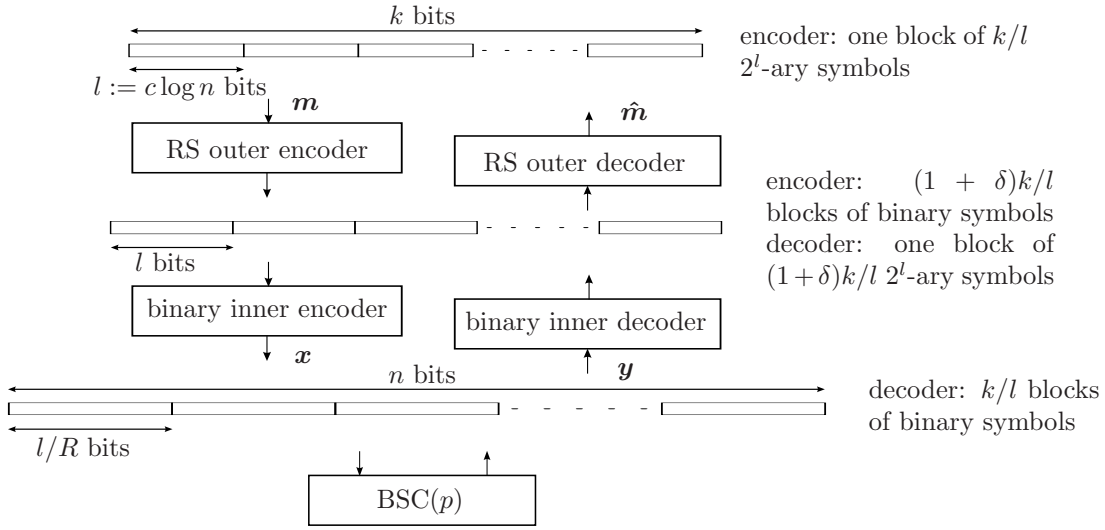


Figure 4: Concationation code

6. The decoder treat the sequence as one block of $(1+\delta)k/l$ 2^l-ary symbols and uses the Reed-Solomon outer code to decode it to an estimate of the original k-bit information sequence.

The Reed-Solomon code can correct up to $\delta k/2l$ errors in the block of $(1+\delta)k/l$ 2^l-ary symbol sequence while the probability of error in a particular block is $2^{-E_r(R)l/R} = n^{-cE_r(R)/R}$. By the union bound, the overall probability of error is upper bounded by $\binom{(1+\delta)k/l}{\delta k/2l}(2^{-E_r(R)l/R})^{\delta k/2l} \approx 2^{-n(E_r(R)\delta/2 - H(\delta/2(1+\delta)))}$.