Lý thuyết thông tin

ptit.edu.vn

Phần 3: Lý thuyết mã hóa

- Toán học nền tảng
- Mã tối ưu
- Mã khống chế lỗi

dinhptit@gmail.com

Toán học nền tảng

- ✓ Số học modular
- ✓ Các cấu trúc đại số:
- ✓ Nhóm
- ✓ Vành
- ✓ Trường
- ✓ Không gian tuyến tính V_n trên GF(2)

Các khái niệm

- **Phép Mã hóa f** là ánh xạ 1- 1 mỗi dấu a_i thành từ mã $\alpha_i^{n_i}$

$$f : a_i \rightarrow \alpha_i^{n_i} \qquad \alpha_i^{n_i} = (b_{i_1}, b_{i_2}, ..., b_{i_{n_i}})$$

n_i : số dấu mã có trong codeword.

  - Mã đều
  - Mã không đều

- **Bộ Mã (code)** là tập các từ mã được dùng để mã hóa các dấu.

$$C = \{\alpha_i^{n_i}\}$$

Các khái niệm(cont)

•Độ chậm giải mã của bộ mã: Là số dấu mã nhận được cần thiết trước khi có thể thực hiện phân tách được từ mã.

•Bộ mã được gọi là không suy biến (nonsingular) nếu mỗi tin của nguồn được mã hóa bằng một từ mã riêng biệt

•Bộ mã phân tách được nếu giải mã là đơn trị (uniquely decodable code) .

•Bộ mã có khả năng giải mã tức thời (Instantaneous code):

- Là mã phân tách được, và
- Sự giải mã được thực hiện ngay trong khi các dẫu mã đang tới mà ko cần đợi tới khi kết thúc nhận bản tin.

→Bộ mã này phải có đặc tính prefix (không có bất cứ từ mã nào là tiền tố của từ mã khác).

Định lý Kraft (về chiều dài các từ mã của bộ mã prefix)

Xét một nguồn tin X={x_1, x_2,…, x_s}.

Điều kiện cần và đủ để tồn tại bộ mã tức thời (prefix) với độ dài tương ứng {n_1, n_2,…,n_s} để mã hóa X là:

$$\sum_{i=1}^{s} m^{-n_i} \leq 1$$

m là cơ số mã.

Độ thừa của một bộ mã đều (D):

Cho nguồn rời rạc A gồm s tin: $A = \{a_i;\ \overline{1,s}\}$.

Xét phép mã hóa *f* : $f : a_i \to \alpha_i^n\ ;\ \alpha_i^n \in V$.

Số từ mã (có độ dài n) có thể có: $N = m^n$. (m: cơ số mã)

Số từ mã được dùng: s

Định nghĩa: Độ thừa của một bộ mã đều:

$$D \overset{\Delta}{=} \frac{H_0(V) - H_0(A)}{H_0(V)} = 1 - \frac{H_0(A)}{H_0(V)}\ [\%]$$

Trong đó : $H_0(A) = \log s$, và $H_0(V) = \log N = n \log m$

- Nếu s=N: Mã không có độ thừa (mã đầy)
- Nếu s<N: Mã vơi (Sẽ có một số tổ hợp mã cấm ko dùng đến)

Khoảng cách mã (d)

Định nghĩa: Khoảng cách giữa hai từ mã bất kỳ α_i^n và α_j^n là số các dấu mã khác nhau tính theo cùng một vị trí giữa hai từ mã này, ký hiệu $d\left(\alpha_i^n, \alpha_j^n\right)$

Tính chất

- $d\left(\alpha_i^n, \alpha_j^n\right) = d\left(\alpha_j^n, \alpha_i^n\right)$
- $n \geq d\left(\alpha_i^n, \alpha_j^n\right) \geq 0$
- (Tính chất tam giác): $d\left(\alpha_i^n, \alpha_j^n\right) + d\left(\alpha_j^n, \alpha_k^n\right) \geq d\left(\alpha_i^n, \alpha_k^n\right)$

Định nghĩa: Khoảng cách Hamming d_0 của một bộ mã được xác định theo biểu thức sau:

$$d_0 \overset{\Delta}{=} \min_{\forall \alpha_i^n, \alpha_j^n} d\left(\alpha_i^n, \alpha_j^n\right)$$

Ở đây α_i^n và α_j^n Là các cặp từ mã phân biệt

Trọng số của một từ mã

Định nghĩa: Trọng số của một từ mã α_i^n (được k/hiệu là $W\left(\alpha_i^n\right)$) là số các dấu mã khác không trong từ mã.

Tính chất của trọng số:

- $0 \le W\left(\alpha_i^n\right) \le$ n
- $d\left(\alpha_i^n, \alpha_j^n\right) = W\left(\alpha_i^n + \alpha_j^n\right)$

Khả năng khống chế lỗi của mã đều nhị phân

- Một mã đều nhị phân có khả năng phát hiện được lỗi nếu số lỗi không vượt quá $\mathrm{d}_0 - 1$.

- Một mã đều nhị phân có khả năng sửa được lỗi nếu số lỗi không vượt quá $t = \left\lfloor \frac{d_0 - 1}{2} \right\rfloor$

Điều kiện sử dụng mã không có độ thừa trên BSC

Giả sử ta truyền từ mã đơn giản qua kênh đối xứng nhị phân không nhớ có xác suất thu sai một dấu là p_0. Khi đó xác suất thu đúng một dấu tương ứng là $(1-p_0)$. Từ mã n dấu chỉ nhận đúng khi mọi dấu mã đều nhận đúng. Như vậy, xác suất thu đúng từ mã $p_đ$ là:

$$p_đ = (1-p_0)^n$$

Xác suất thu sai của từ mã là:

$$p_s = 1 - p_đ = 1 - (1-p_0)^n$$

Nếu xác suất thu sai cho phép một từ mã (n dấu) là p_{scp}, khi đó điều kiện sử dụng mã đơn giản trong kênh nhị phân đối xứng ko nhớ là:

$$p_s \le p_{scp}$$

Với $p_0 << 1$, ta có công thức gần đúng sau:

$$(1-p_0)^n \approx 1 - n p_0$$

Do đó: $p_s \approx n p_0$

khi đó điều kiện sử dụng mã đơn giản trên BSC: $p_0 \le \frac{p_{scp}}{n}$

ptit.edu.vn

Mã tối ưu

dinhptit@gmail.com

Mã hóa nguồn

Xét nguồn tin $A = \begin{pmatrix} a_i \\ p(a_i) \end{pmatrix}$, $i = \overline{1,S}$

Phép mã hóa nguồn $f : \ a_i \to \alpha_i^{n_i}$

- **Mục tiêu của mã nguồn**

  Nén (giảm độ dài trung bình của từ mã)

- **Độ dài trung bình của bộ mã**

  Là kỳ vọng của đại lượng ngẫu nhiên n_i.

$$\overline{n} = M[n_i] = \sum_{i=1}^{s} n_i p(a_i)$$

Định lý mã hóa nguồn của Shannon (Định lý mã hóa nguồn 1)

- Độ dài trung bình từ mã của bất cứ bộ mã prefix nào dùng để mã hóa nguồn tin A cũng không thể để bé hơn n_0 , tức là:

$$\bar{n} \geq n_0 \qquad n_0 = \frac{H(A)}{\log m}$$

m: Cơ số mã.

Dấu "=" xảy ra khi: $m^{-n_i} = p(a_i)$

- Nguyên tắc mã nguồn là ưu tiên từ mã ngắn để mã hóa các tin có xác suất xuất hiện lớn.
- Hiệu quả của phép mã hóa nguồn: $\eta = \frac{n_0}{\bar{n}}$

Một số thuật toán mã nguồn

- **Thuật toán Shannon**
  - Thuộc lớp mã entropy.
- **Thuật toán Shannon-Fano**
  - Thuộc lớp mã entropy. Thuật toán đơn giản, Bộ mã có tính prefix
  - Không luôn tạo được mã tối ưu, Mã cận tối ưu (Suboptimal)
- **Thuật toán Shannon-Fano-Elias**
  - Là tiền thân của mã hóa số học. Ít dùng vì có $H(x)+1 \leq LC(X) \leq H(X)+2$
- **Thuật toán số học**
  - Thuộc lớp mã entropy, là sự mở rộng thuật toán Shannon-Fano-Elias.
- **Thuật toán Lempel-Ziv**
  - Mã hóa từ điển. Thuật toán thích nghi. Không thuộc lớp entropy.
  - Không yêu cầu phải biết trước phân bố của nguồn.
- **Thuật toán Huffman**

Mã tối ưu

- **Yêu cầu của Optimal Codes**
  - Bộ mã phải có khả năng giải mã tức thời (tính prefix).
  - Độ dài trung bình từ mã phải đạt giá trị tối ưu $(\overline{n} = \overline{n}_{opt})$

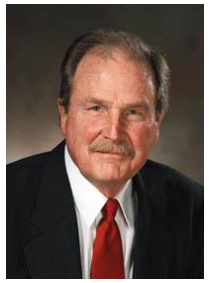
- **Bất đẳng thức kẹp (Định lý mã hóa nguồn 2)**

  Độ dài trung bình của một bộ mã tối ưu thỏa mãn:

$$n_0 \leq \overline{n}_{opt} < n_0 + 1$$

Thuật toán Huffman

Đề xuất bởi David A. Huffman:



Đặc điểm

- Mã hóa entropy dựa trên tần suất xuất hiện
- Nén lossless;
- Variable-length code
- Bộ mã thu được là opt: (có tính prefix, và có $(\overline{n} = \overline{n}_{opt})$

Nhược điểm

- Phải biết trước phân bố của nguồn
- Bộ mã ko phải là duy nhất, vì vậy phải gửi kèm cây mã (bảng mã) để phục vụ việc giải mã sau này.

ptit.edu.vn

Channel Error Control

dinhptit@gmail.com

Overview

❑ The need for Error Control

- **Ideal channel**: Sự truyền tin ko bị **corruption & error**.
- **Real channel:** BW, Noise, distortion & interference**,** gây ra lỗi

❑ Principles of Error Control

Phía phát: Tạo ra redundant bits vào chuỗi digit nhờ thuật toán, sao cho các lỗi sinh ra trên kênh có thể kiểm soát tại máy thu .

❑ Error Control Schemes

- ARQ schemes (Automatic Repeat request)
- FEC schemes (Forward Error Correction):
- Hybrid schemes

❑ Techniques for error detection

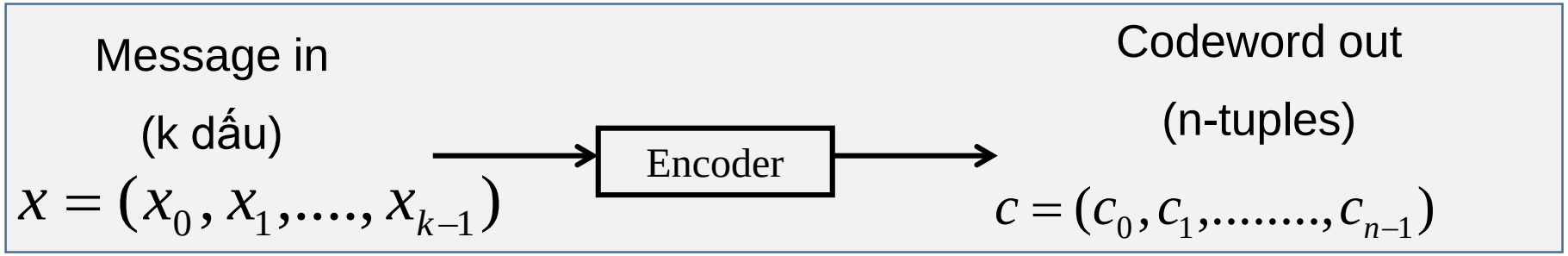
- Parity check
- Checksum
- Cyclic redundancy check (CRC)

Error-correcting codes (ECC)

- Linear block codes
  - ✓ Cyclic codes (e.g., Hamming codes is a subset)
  - ✓ Repetition codes
  - ✓ Polynomial codes (e.g., BCH codes)
  - ✓ Reed–Solomon codes
  - ✓ Low-density parity-check (LDPC) codes
- Convolutional codes
- Tubo codes

Mã khối (block codes)

- **Mã hóa khối:** là những thuật toán mã hóa hoạt động trên những khối thông tin vào có độ dài k xác định, tạo ra các từ mã có độ dài n xác định.



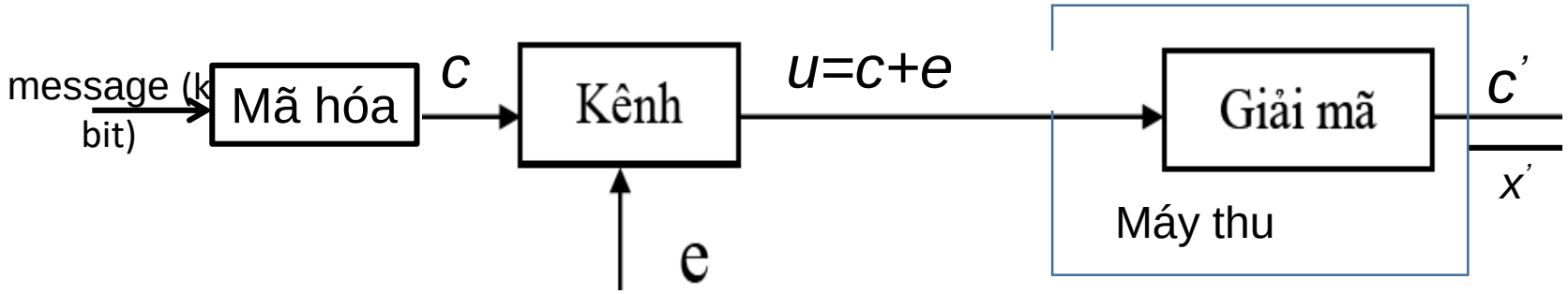
$\cdot\ x_i \in GF(q);\quad c_j \in GF(q)$

q: Cơ số mã, mặc định là 2.

- **Bộ mã khối C(n,k):** Là tập gồm 2^k từ mã n-tuples, được chọn trong số 2^n tổ hợp mã có thể có độ dài n.

- **Code rate :** $R_c = \frac{k}{n}$

Mô hình kênh có nhiễu cộng:



$c = (c_0, c_1, ..., c_{n-1})$: Từ mã phát (n-tuples), ϵ bộ mã C(n,k)

$u = (u_0, u_1, ..., u_{n-1})$: Vector thu.

$e = (e_0, e_1, ..., e_{n-1})$:Mẫu lỗi, → có 2^n cấu trúc lỗi.

- Error partten $e = u + c = (u_0 + c_0, u_1 + c_1, ..., u_{n-1} + c_{n-1}) = (e_0, e_1, ..., e_{n-1})$

$$where\ e_i = 1\ \text{for}\ u_i \neq c_i, \text{and}\ e_i = 0\ \text{for}\ u_i = c_i$$

- Máy thu không phát hiện ra có lỗi nếu vector thu u là một từ mã hợp lệ.

Khả năng khống chế lỗi của mã khối

- Một bộ mã (n,k,d_0) có khả năng phát hiện được lỗi nếu cấu trúc lỗi có trọng số thỏa mãn:

$$w(e) \le (d_0 - 1)$$

- Một bộ mã (n,k,d_0) có khả năng tự sửa được lỗi nếu cấu trúc lỗi có trọng số thỏa mãn:

$$w(e) \le \left\lfloor \frac{d_0 - 1}{2} \right\rfloor$$

ptit.edu.vn

Mã khối tuyến tính (Linear block codes)

dinhptit@gmail.com

Mã khối tuyến tính (linear block code)

Định nghĩa: Mã khối tuyến tính C(n,k) có chiều dài từ mã n, trong đó mỗi dấu mã là một dạng tuyến tính của k dấu thông tin.

$$c = (c_0 c_1 \ldots\ldots c_{n-1}) \qquad c_j = \sum_{i=1}^{k} a_i x_i \qquad j = 0,\ldots,n-1$$

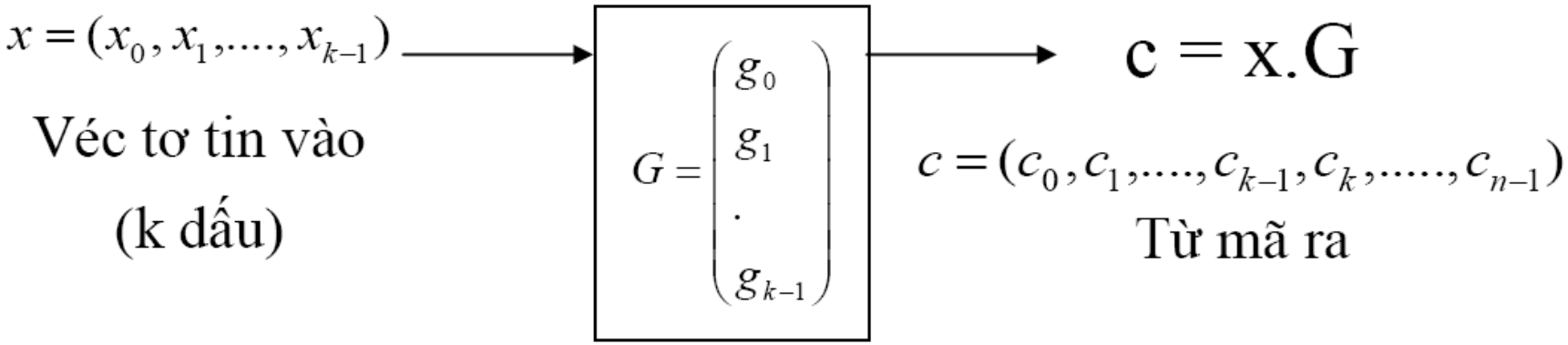
Định nghĩa: Mã tuyến tính (n,k) là không gian con tuyến tính k chiều (V_k) trong không gian tuyến tính n chiều (V_n).

❑ Generator matrix của mã khối tuyến tính (n,k):

$$G = \begin{pmatrix} g_0 \\ g_1 \\ \ldots \\ g_{k-1} \end{pmatrix} = \begin{pmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,n-1} \\ \cdots & \cdots & \cdots & \cdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{pmatrix}$$

Mã khối tuyến tính (tt)

❑ Mô hình tạo mã khối tuyến tính (n,k):



❑ Ma trận kiểm tra

Mỗi mã C(n,k), tồn tại mã đối ngẫu C_d(n,n-k) với ma trận sinh H.

$$H = \begin{pmatrix} h_0 \\ h_1 \\ \dots \\ h_{n-k-1} \end{pmatrix} = \begin{pmatrix} h_{0,0} & h_{0,1} & \dots & h_{0,n-1} \\ h_{1,0} & h_{1,1} & \dots & h_{1,n-1} \\ \dots & \dots & \dots & \dots \\ h_{n-k-1,0} & h_{n-k-1,1} & \dots & h_{n-k-1,n-1} \end{pmatrix}$$

H is called a *parity-check* matrix of *C*.

Mã hệ thống tuyến tính

Dạng thức 1:

Redundant checking part	message part
(n-k) digits	k digits

$$G_{sys} = \begin{pmatrix} g_0 \\ g_1 \\ \dots \\ g_{k-1} \end{pmatrix} = \begin{pmatrix} p_{0,0} & \dots & p_{0,n-k-1} & I_{0,n-k} & \dots & I_{0,n-1} \\ p_{1,0} & \dots & p_{1,n-k-1} & I_{1,n-k} & \dots & I_{1,n-1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ p_{k-1,0} & \dots & p_{k-1,n-k-1} & I_{k-1,n-k} & \dots & gI_{k-1,n-1} \end{pmatrix} = [PI_k]$$

Matrix $P_{k \times (n-k)}$ Matrix I_k

- Tương ứng các dấu mã kiểm tra và các dấu mã mang thông tin:

$$c_j = x_0 p_{0,j} + x_1 p_{1,j} + \dots + x_{k-1} p_{k-1,j} \qquad 0 \le j \le n-k-1$$

$$c_{n-k+i} = x_i \qquad 0 \le i \le k-1$$

- Parity-check matrix in systematic form is: $H_{sys} = \begin{pmatrix} h_0 \\ h_1 \\ \dots \\ h_{n-k-1} \end{pmatrix} = [I_{n-k} P^T]$

Mã hệ thống tuyến tính (tt)

Dạng thức 2:

message part	Redundant checking part
k digits	(n-k) digits

- Generator matrix in systematic form is:

$$G_{sys} = \begin{pmatrix} g_0 \\ g_1 \\ \dots \\ g_{k-1} \end{pmatrix} = \begin{pmatrix} I_{0,0} & \dots & I_{0,k-1} & p_{0,k} & \dots & p_{0,n-1} \\ I_{1,0} & \dots & I_{1,k-1} & p_{1,k} & \dots & p_{1,n-1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ I_{k-1,0} & \dots & I_{k-1,k-1} & p_{k-1,k} & \dots & p_{k-1,n-1} \end{pmatrix} = [I_k P]$$

Ma trận $\mathbf{I_k}$ Ma trận $\mathbf{P_{k\ x\ (n-k)}}$

- Tương ứng các dấu mã mang thông tin và các dấu kiểm tra trong từ mã:

$c_i = x_i \quad 0 \le i \le k\text{-}1; \quad c_j = x_0 p_{0,j} + x_1 p_{1,j} + \dots + x_{k-1} p_{k-1,j} \quad k \le j \le n\text{-}1$

- Parity-check matrix in systematic form is:

$$H_{sys} = \begin{pmatrix} h_0 \\ h_1 \\ \dots \\ h_{n-k-1} \end{pmatrix} = [P^T I_{n-k}]$$

Các bài toán tối ưu của mã tuyến tính nhị phân

Khi xây dựng một mã tuyến tính (n,k,d_0) người ta mong muốn tìm được các mã có độ thừa nhỏ nhưng lại có khả năng khống chế sai lớn. Để đơn giản người ta thường xây dựng mã dựa trên các bài toán tối ưu sau:

Bài toán 1

Cho trước k và d_0, tìm mã có độ dài với từ mã n nhỏ nhất.

Tương ứng với bài toán này ta có giới hạn Griesmer sau:

$$n \geq \sum_{i=0}^{k-1} \left\lceil \frac{d_0}{2^i} \right\rceil$$

Ở đây $\lceil x \rceil$ chỉ số nguyên nhỏ nhất lớn hơn hoặc bằng *x*.

Ví dụ Cho $k=4,\ d_0=3$

$$n \geq 3+2+1+1=7$$

Hay nói một cách khác mã (7, 4, 3) là một mã tối ưu đạt được giới hạn Griesmer.

Các bài toán tối ưu của mã tuyến tính nhị phân (tt)

Bài toán 2

Cho n và k, tìm mã có d_0 là lớn nhất.

Tương ứng với bài toán này ta có giới hạn Plotkin sau:

$$d_0 \le \frac{n.2^{k-1}}{2^k - 1}$$

Ví dụ: Cho k = 3, n = 7

$$d_0 \le \frac{7.2^2}{2^3 - 1} = 4$$

Nói một cách khác mã (7, 3, 4) là một mã tối ưu đạt được giới hạn Plotkin

Các bài toán tối ưu của mã tuyến tính nhị phân (tt)

Bài toán 3

Cho n và số sai khả sửa t xác định (hoặc cho n, d_0), tìm mã có số dấu thông tin k là lớn nhất (hay số dấu thừa $r = n - k$ là nhỏ nhất)

Tương ứng với bài toán này ta có giới hạn Hamming sau:

$$2^{n-k} \geq \sum_{i=0}^{t} C_n^i$$

Ví dụ Cho n = 7 và t = 1

$$2^r \geq \sum_{i=0}^{1} C_7^i = C_7^0 + C_7^1 = 8$$

$$r \geq \log_2 8 = 3$$

hay $k \leq 7 - 3 = 4$

Như vậy mã (7, 4, 3) là mã tối ưu đạt được giới hạn Hamming

Mã đạt được giới hạn Hamming còn được gọi là mã hoàn thiện

Tài liệu tham khảo

- ✓ John Proakis & Masoud Salehi, **Digital Communication,** 2007
- ✓ Shu Lin, **Error Control Coding-Fundamentals and Aplications**, Prentice Hall, 2004
- ✓ Simon Haykin, **Communication Systems**, 4rd edition, John Wiley & Sons, 2001.