L05

Wednesday, August 31, 2022 1:11 PM

Constructing Capacity-achieving ECCs.

* Binary codes:

  each codeword (a point in the high-dim space) $\vec{x} \in \{0,1\}^n$

  $\{\vec{x}$: all codewords$\}$ is termed the codebook.

* Binary Random codes:

  Fix the design parameter $P_X(x) = \begin{cases} p & \text{if } x=0 \\ 1-p & \text{if } x=1 \end{cases}$

  $\vec{x}_1 = (x_{1,1}, x_{1,2}, \cdots, x_{1,n})$

  $\vdots$

  $\vec{x}_{2^{rn}} = (x_{2^{rn},1}, \cdots, x_{2^{rn},n})$

  We have 2^{rn} distinct codewords, each

We have 2^{rn} distinct codewords, Each entry is chosen w. i.i.d. P_X

r: Code rate (bits/symbol usage

n: codeword length

① Tractable Analysis

That is if $n \to \infty$ and the random code has $r < I_{P_X}(X;Y)$

then the error-rate $\to 0$

② We can choose arbitrary P_X

ex: For Z-channel,

$$P_X(x) = \begin{cases} \frac{1-\varepsilon^{\frac{1}{1-\varepsilon}}}{1+(1-\varepsilon)\varepsilon^{\frac{\varepsilon}{1-\varepsilon}}} & \text{if } x=0 \\ \frac{\varepsilon^{\frac{\varepsilon}{1-\varepsilon}}}{1+(1-\varepsilon)\varepsilon^{\frac{\varepsilon}{1-\varepsilon}}} & \text{if } x=1 \end{cases}$$

③ Not practical for long n

Encoding complexity: $O(2^{rn})$

Encoding complexity: $O(2^{rn})$

Decoding complexity: $O(2^{rn})$

Binary linear codes

Definition 1: $\{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_{2^{rn}}\}$ form a binary linear subspace of $(GF(2))^n$ where $GF(2)$ is the binary field.

Definition 2: $\exists$ a binary $n \times rn$ generating matrix G s.t. any codeword $\vec{x}$ (column vector) is expressed

$\vec{x} = G\vec{m}$ where $\vec{m} \in (GF(2))^{rn}$ is the information bit string of length rn

Encoding Complexity: $O(n^2)$ matrix multiplication

multiplication

① $\vec{x}, \vec{m}$ are column vectors.

② All operations are in the binary field:

$1+1=0$ $\quad 1\cdot 1=1$

$0+1=1$ $\quad 1\cdot 0=0$

$0+0=0$

Lemma 1: Assuming the "message vector" $\vec{m}$ is uniformly distributed over $[GF(2)]^m$.

For any linear code, consider the marginal distribution of the ith bit X_i

$\Rightarrow P_{X_i}$ is Bernoulli distributed with either $P(X_i=1)=\frac{1}{2}$ or $P(X_i=1)=0$ #

Proof: For any linear code, consider its generating matrix representation

$$\vec{x} = G\vec{m} = (\vec{g}_1, \vec{g}_2, \cdots \vec{g}_m)\vec{m}$$

Consider the i-th bit x_i, & use $g_{i,j}$ to denote the i-th coordinate of the j-th vector $\vec{g}_j$

Case 1: If one of g_{ij} is non-zero, (say $g_{i,j_0}=1$, then

$$x_i = m_{j_0} + (g_{i1}, g_{i,2}, \cdots g_{i,j_0-1}, g_{i,j_0+1}, \cdots g_{ik}) \cdot (m_1, \cdots, m_{j_0-1}, m_{j_0+1}, \cdots, m_{rn})$$

Since m_{j_0} is uniform Bernoulli, so is X_i

Case 2: All $g_{i,j}$ are zero.

Then $x_i = (0, \cdots, 0) \cdot (m_1, \cdots, m_{rn}) = 0$ ✗

✗ Corollary: Linear codes do not achieve the capacity of a z-channel, which requires non-uniform P_X.

✗ Fortunately, linear codes can achieve most capacity when choosing each entry of G uniformly randomly.

Definition 3: ∃ a $(1-r)n \times n$ parity check matrix H s.t. $H \vec{x} = \vec{0}$ holds

Check matrix H s.t. $\forall \vec{x}$ being a codeword, we have

$$H\vec{x} = 0$$

Example: Hamming code

$n = 7$, $rn = 4$ (or $k = 4$)

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

ex: $\vec{x} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$ is a codeword

Total # of codewords: $2^{n - \text{RowRank}(H)}$

Converting between G & H

H —row operation→ $H' = (I_3 \; P)$

$$G = \begin{pmatrix} P \\ I_4 \end{pmatrix}$$

$\Rightarrow H'G = 0$

$\Rightarrow H'G = 0$

$\Rightarrow G = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$

Now we can index each codeword $\vec{x}$

$\vec{x}_0 = \vec{x}_{0000} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

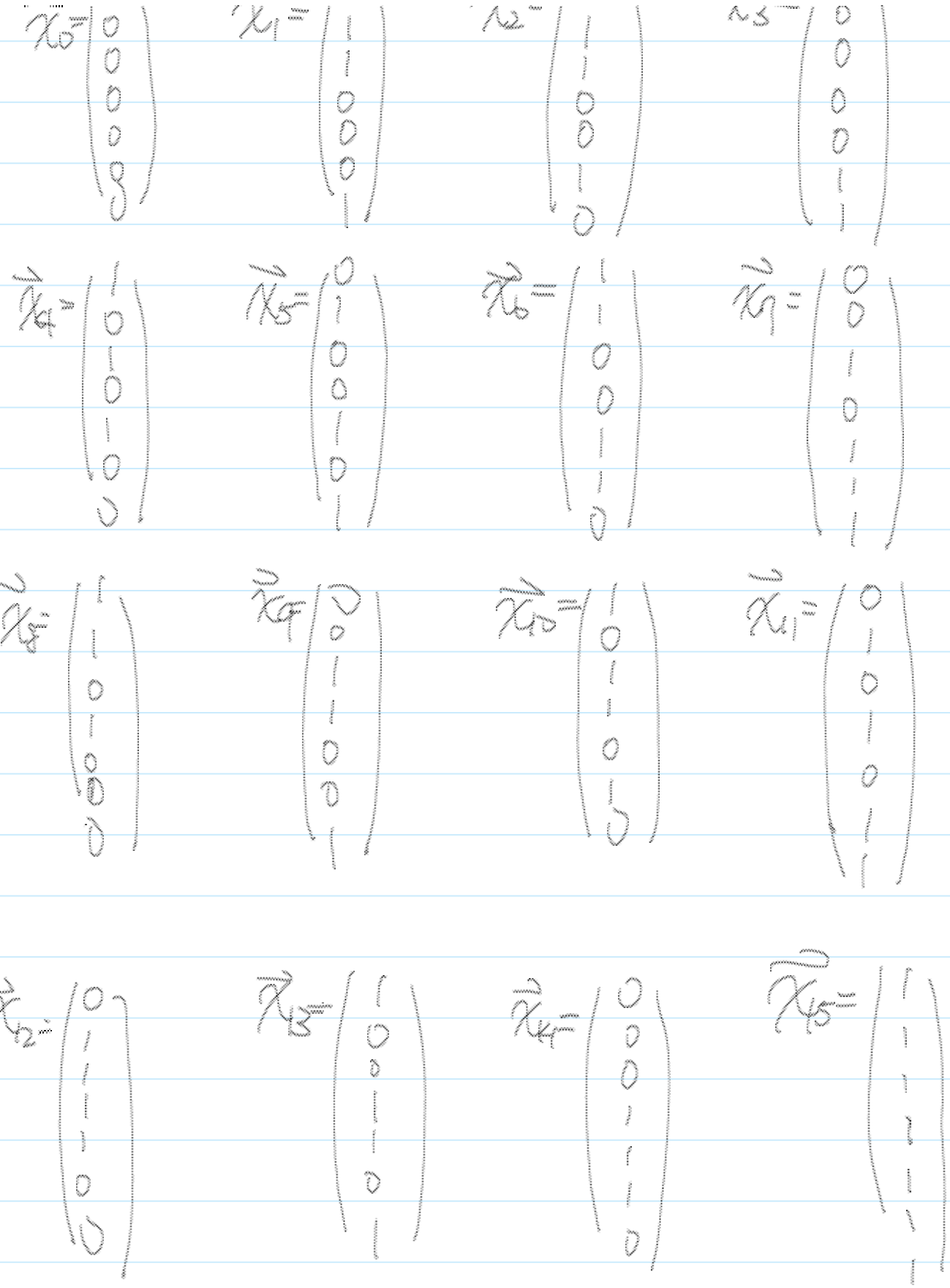
$\vec{x}_1 = \vec{x}_{0001} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$

$\vec{x}_2 = \vec{x}_{0010} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

$\vec{x}_{14} = \vec{x}_{1110} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$

$\vec{x}_{15} = \vec{x}_{1111} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$

$\vec{x}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ $\vec{x}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ $\vec{x}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ $\vec{x}_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

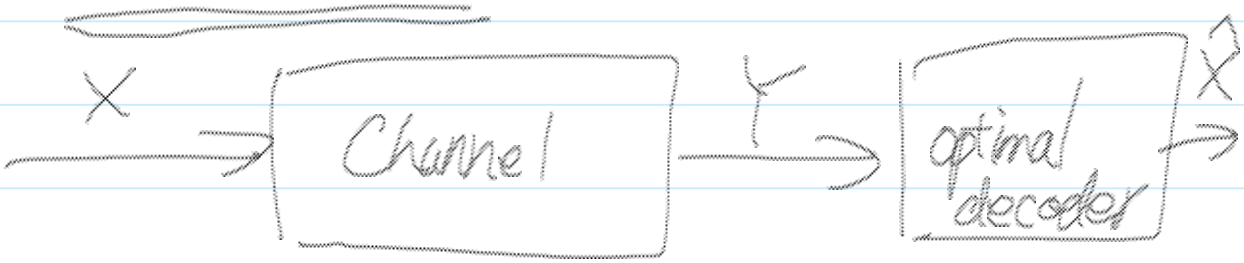


Q: What is the optimal decoder?

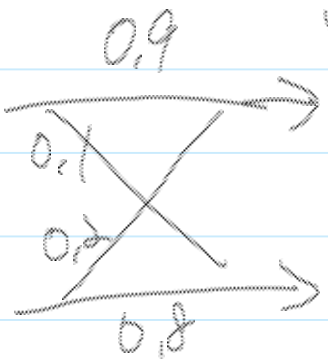
Syndrome decoder, Error-Trapping decoder, ... ? Which is optimal (minizing the decoding error.)

Or the minimal distance decoder.

* The answer depends on the channel model.



Example: The i.i.d channel follows



for each x_i

Q: What is the optimal decoder?

$\hat{X}(\vec{y})$ Say when the observation is

$\hat{\vec{x}}(\vec{y})$ say when the observation is $\vec{y} = (1\,1\,1\,0\,1\,1\,1\,0)$

Ans: It is no different than a hypothesis testing problem with 16 competing candidates.

H_0: $Y_1, \ldots, Y_n$ follow $p_{\vec{Y}|\vec{X}}(\cdot\,|\,\vec{x}_0)$

H_1

$\vdots$

H_{15} $p_{\vec{Y}|\vec{X}}(\cdot\,|\,\vec{x}_{15})$

with $P(\vec{X} = \vec{x}_i) = \frac{1}{16}$ for $i = 0, \ldots, 15$

The optimal decoder is simply the

$\hat{\vec{x}}_{MAP}(\vec{y})$ MAP decoder

(or the ML decoder since $P(\vec{X} = \vec{x}_i)$ is uniform

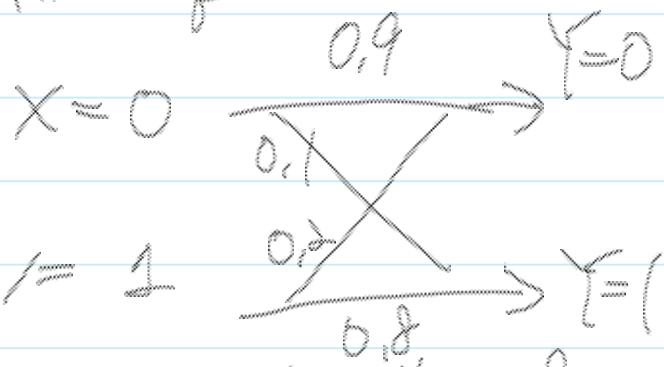
The 16 likelihood values are

$L_0 = (0.1)^5 \times 0.9^2$ $P_{Y|\vec{X}}(1110110 \mid 0000000)$

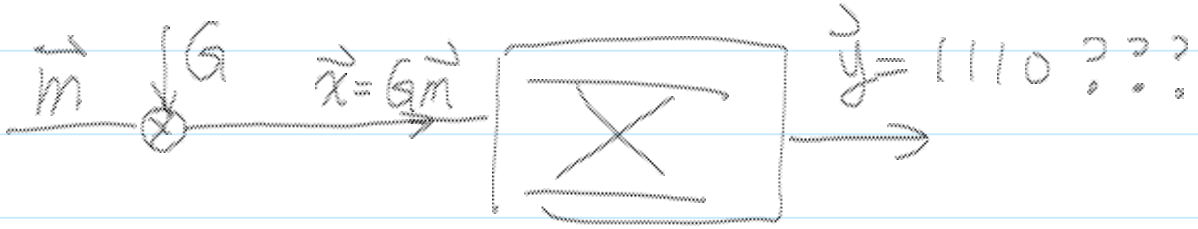
$L_1 = 0.1^2 \times 0.9 \times 0.8^3 \times 0.2$ $P_{Y|\vec{X}}(1110110 \mid 1110001)$

$L_2 = 0.1^2 \times 0.9^2 \times 0.8^3$ $P_{Y|\vec{X}}(1110110 \mid 0110010)$

Q: Assume the same channel model as in the first question:



Suppose now only the first 4 bits can be observed



Find the most likely codeword:

The 16 likelihood values are

$L_0 = (0.1)^3 \times 0.9$ $\quad P_{Y|X}(1110\square\square\square \mid 0000000)$

$L_1 = 0.8^3 \times 0.9$ $\quad P_{Y|X}(1110\square\square\square \mid 1110001)$

$L_2 = 0.1 \times (0.8)^2 \times 0.9$ $\quad P_{Y|X}(1110\square\square\square \mid 0110010)$

$\vdots$

$$\hat{x}_{MAP}(\vec{y}) = \arg\max_{\vec{x}_m} P_{\vec{X}|\vec{Y}}(\vec{x}_m \mid \vec{y})$$

$$= \vec{x}_1$$

Q: Suppose $\vec{y}_{obs} = (1,1,1,0,1,1,0)$ & the same channel model.

What is the optimal MAP decoder for the 2nd bit of $\vec{x}$?

This is the optimal "bit decoder" different from the optimal "codeword/frame decoder"

Ans: $\vec{X} = (X_1, X_2, \cdots, X_7)$

$$P_{X_2|\vec{Y}}(0|\vec{y}_{obs}) = \frac{\sum_{\vec{x}:\text{ codeword } \& \ x_2=0} P(\vec{X}=\vec{x}) \cdot P_{\vec{Y}|\vec{X}}(\vec{y}_{obs}|\vec{x})}{\sum_{\vec{x}:\text{ codeword}} P(\vec{X}=\vec{x}) \cdot P_{\vec{Y}|\vec{X}}(\vec{y}_{obs}|\vec{x})}$$

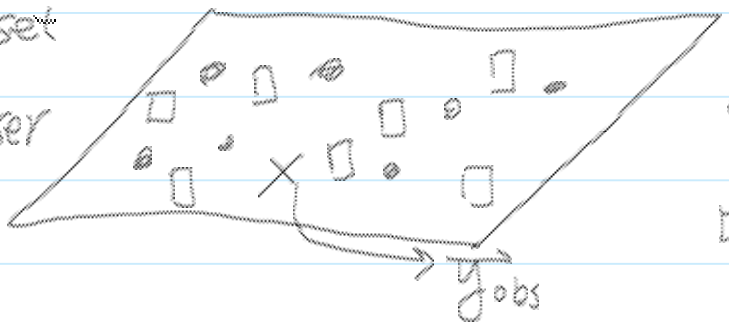
$$= \frac{\frac{1}{16} \times \sum_{m=0,3,4,7,9,10,13,14} P_{\vec{Y}|\vec{X}}(\vec{y}_{obs}|\vec{x}_m)}{\frac{1}{16} \sum_{m=0}^{15} P_{\vec{Y}|\vec{X}}(\vec{y}_{obs}|\vec{x}_m)}$$

$$< \frac{1}{2} \Rightarrow \hat{X}_{2,MAP}(\vec{y}) = 1$$

$$\hat{X}_{2,MAP}(\vec{y}_{obs}) = \underset{x=0,1}{\arg\max} \; P_{X_2|\vec{Y}}(x|\vec{y}_{obs})$$

Again, $P(\hat{X}_{2,MAP}(\vec{Y}) \neq X_2)$ is minimized.

In prob, which set of codewords is closer



In summary:

In summary:

$\hat{x}_{i,MAP}(\vec{y})$: the optimal bit decoder of x_i that minimizes the i-th bit error probability (bit error rate)

$\hat{x}_{MAP}(\vec{y})$: the optimal codeword detector that minimizes the codeword error prob. (frame error rate)

Q:

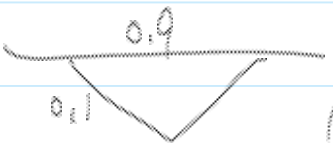
$$\hat{x}_{MAP,bitwise}(\vec{y}) = \left(\hat{x}_{1,MAP}(\vec{y}), \hat{x}_{2,MAP}(\vec{y}), \cdots, \hat{x}_{n,MAP}(\vec{y})\right)$$

minimizes $\frac{1}{n}\sum_{i=1}^{n} P\left(f(\vec{y})_i \neq x_i\right)$

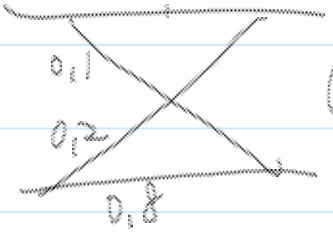
the averaged bit error rate

Exercise: $\vec{y}_{obs} = (1\ 1\ 1\ 0\ 1\ 1\ 0)$

In the channel is



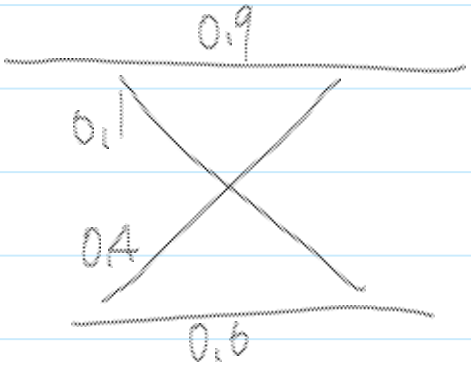
Q: Find $\hat{x}_{MAP}(\vec{y}_{obs})$



Q: Find $\vec{x}_{MAP,bit}(\vec{y}_{obs})$

Exercise: With the same $\vec{y}_{obs}$

& the channel is



Q: Find $\vec{x}_{MAP,bitwise}(\vec{y}_{obs})$

Ans: $\vec{x}_{MAP,bitwise}(\vec{y}_{obs}) = (1\ 1\ 1\ 0\ 1\ 0)$

NOT EVEN A CODEWORD!!

But it is guaranteed to minimize the bit error rate.

Thus far, we have learned how to develop the optimal decoder $\hat{X}_{MAP}(\vec{y})$.

We can also design some near optimal decoders. For example: The minimum distance decoder

$$\hat{X}_{mD}(\vec{y}) = \underset{\vec{x}_m}{\arg\max} \; |\vec{x}_m - \vec{y}|$$

Hamming distance

* The natural question is thus how to evaluate the performance of these decoders?

* Essentially, how to compute

Frame error rate $P(f(\vec{y}) \neq X)$

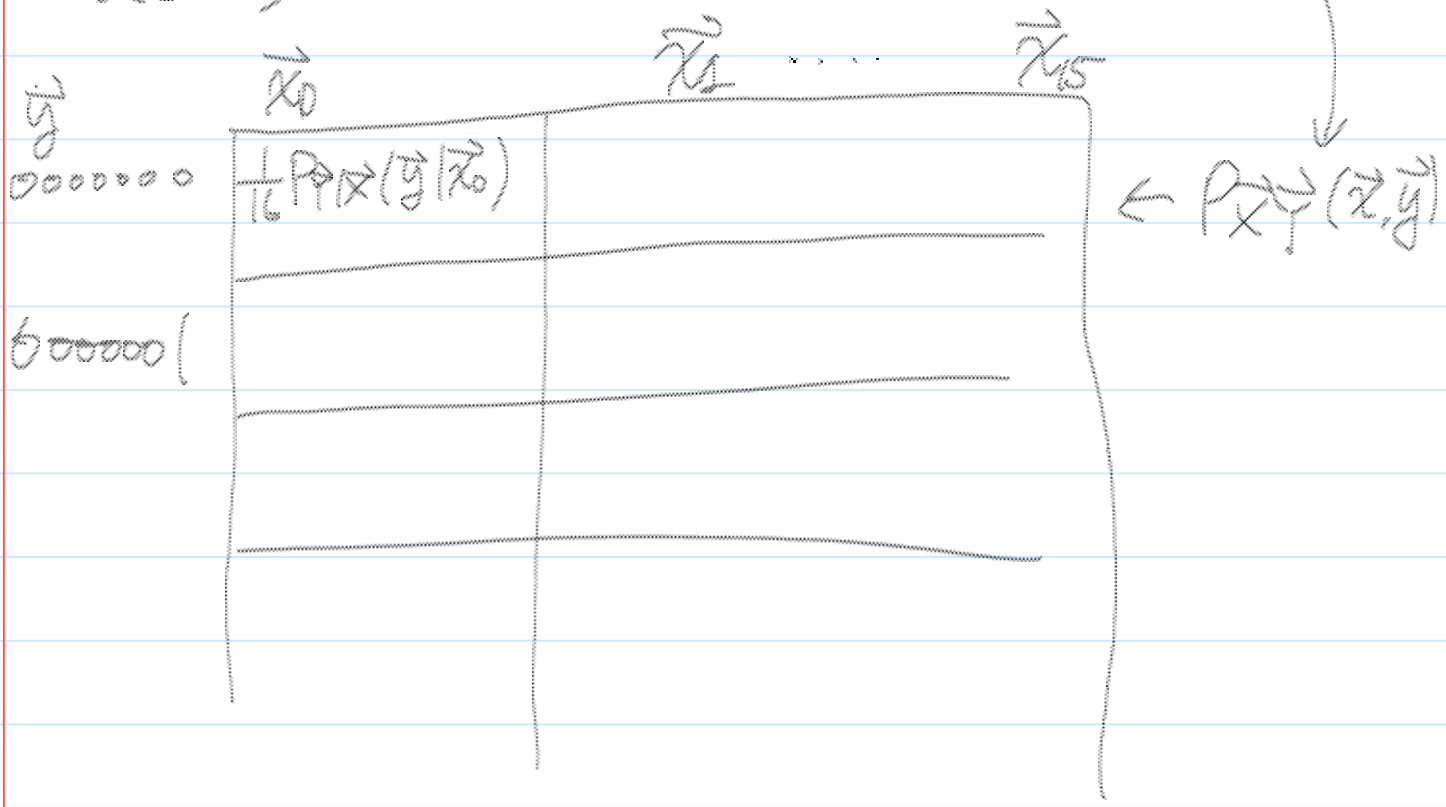
Bit error rate $\frac{1}{n}\sum_{i=1}^{n} P(f(\vec{y})|_i \neq X_i)$

Answer: This problem is no different than the

error probability computation of hypothesis testing.

One just has to count the probability carefully with 16 competing hypothesis

For Hamming code, the joint prob table becomes



* Frame error rate (FER) or word error rate (WER)

P(the decoded vector is not the transmitted codeword)

codeword)

$$= P(f(\vec{Y}) \neq \vec{X})$$

$$FER = \sum_{\vec{y}=000\ldots0}^{111\ldots1} \sum_{m=0}^{15} P_{\vec{X},\vec{Y}}(\vec{x}_m, \vec{y}) \cdot 1\{f(\vec{y}) \neq \vec{x}_m\}$$

where $1\{\neq\} = \begin{cases} 1 & \text{if } \neq \text{ holds} \\ 0 & \text{if not.} \end{cases}$

Bit error rate of the i-th bit.

$$P(f(\vec{Y})|_i \neq X_i)$$

$$*BER_i = \sum_{\vec{y}=0000\ldots0}^{111\ldots1} \sum_{m=0}^{15} P_{\vec{X},\vec{Y}}(\vec{x}, \vec{y}) 1\{f(\vec{y})|_i \neq x_i\}$$

Bit error rate

$$BER = \frac{1}{n} \sum_{i=1}^{n} P(f(\vec{Y})|_i \neq X_i)$$

Alternatively

$$= \frac{1}{n} \sum_{\vec{y}} \sum_{\text{all } \vec{x}_m} P_{\vec{X},\vec{Y}}(\vec{x}, \vec{y}) \cdot \underbrace{|f(\vec{y}) - \vec{x}_m|}_{\text{Hamming distance.}}$$

Handout Binary Symmetric channel with crossover prob p

It is very cumbersome to perform optimal decoding, let alone evaluate its performance.

Usually can be done only for small codes

$\Leftarrow$ Unless we focus on codes of special structure.