Lecture 07.

Note Title 2/1/2012

* Binary Linear Codes.

$\vec{x} = G\vec{m}$ or $H\vec{x} = \vec{0}$

* Binary linear codes could achieve the capacity for $P_X = (\frac{1}{2}, \frac{1}{2})$.

That is, we choose each entry of G equally likely from $\{0, 1\}$.

* Hamming code.

$\vec{x}_0, \cdots, \vec{x}_{15}$

* Optimal decoder

$$\hat{X}_{MAP}(\vec{y}_{obs}) = \operatorname{argmax}_{\vec{x}} P_{Y|X}(\vec{y}_{obs} \mid \vec{x})$$

$\vec{x}$: being a codeword

which minimizes the "frame error rate"

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

(also known as codeword error rates.

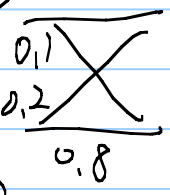
$\vec{x}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ $\vec{x}_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$ $\vec{x}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$ $\vec{x}_3 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$

$\vec{x}_4 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$ $\vec{x}_5 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$ $\vec{x}_6 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$ $\vec{x}_7 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$

$\vec{x}_8 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$ $\vec{x}_9 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$ $\vec{x}_{10} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$ $\vec{x}_{11} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$

$\vec{x}_{12} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$ $\vec{x}_{13} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$ $\vec{x}_{14} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$ $\vec{x}_{15} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$

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

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

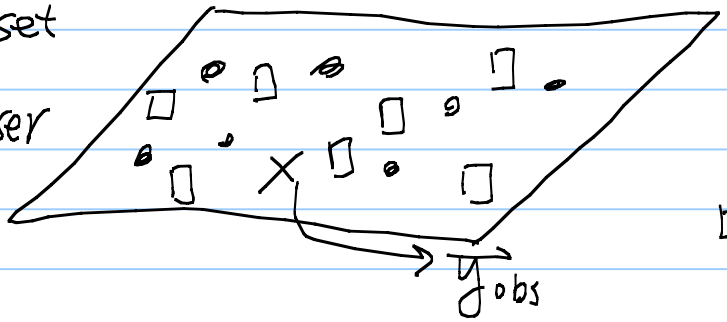
$$= \frac{\frac{1}{16} \times \sum_{m=0,3,4,7,9,10,13,14} P_{\vec{Y}|\vec{X}}(\vec{y}_{obs}|\vec{x}_m)}{\frac{1}{16}\sum_{m=0}^{15} P_{\vec{Y}|\vec{X}}(\vec{y}_{obs}|\vec{x}_m)}$$

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

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

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

In prob, which set of codewords is closer



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:

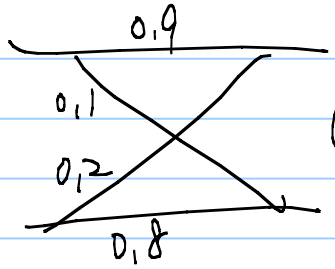
$$\hat{X}_{MAP,\,bitwise}(\vec{y}) = \left(\hat{X}_{1,MAP}(\vec{y}), \hat{X}_{2,MAP}(\vec{y}), \cdots, \hat{X}_{n,MAP}(\vec{y})\right)$$

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

the averaged bit error rate

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

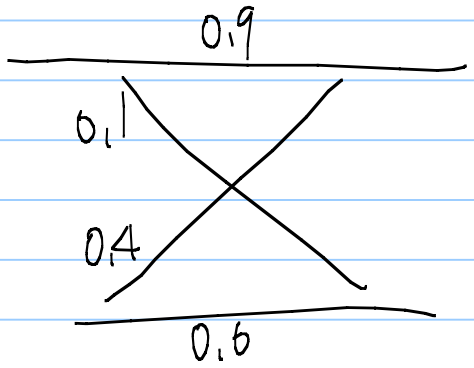
& the channel is



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

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 <u>minimum distance</u> decoder $\hat{X}_{mD}(\vec{y}) = \underset{\vec{x}_m}{\arg\max} \left| \vec{x}_m - \vec{y} \right|$

Hamming distance

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

* Essentially, how to compute

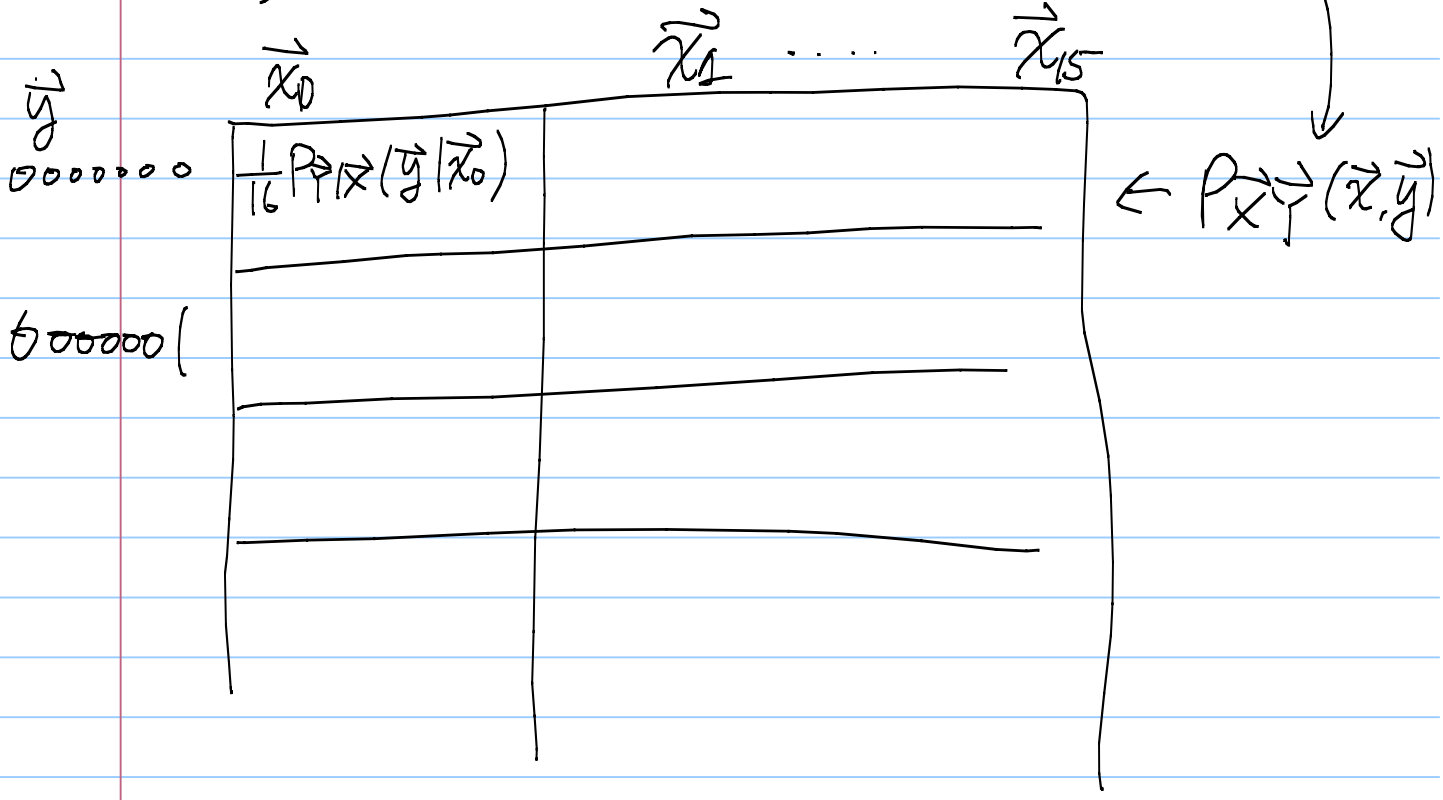
Frame error rate $P(f(\vec{Y}) \neq X)$

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)

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

$$FER = \sum_{\vec{y}=0000000}^{1111111} \sum_{m=0}^{15} P_{\vec{X},\vec{Y}}(\vec{x}_m, \vec{y}) \cdot 1\{f(\vec{y}) \neq \vec{x}_m\}$$

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.

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

$$* BER_i = \sum_{\vec{y}=0000000}^{1111111} \sum_{m=0}^{15} P_{\vec{X}\vec{Y}}(\vec{x}, \vec{y}) 1\{f(\vec{y})|_i \neq x_i\}$$

Bit error rate

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

Alternatively

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

Hamming distance.

Handout Binary symmetric channel with crossover prob p

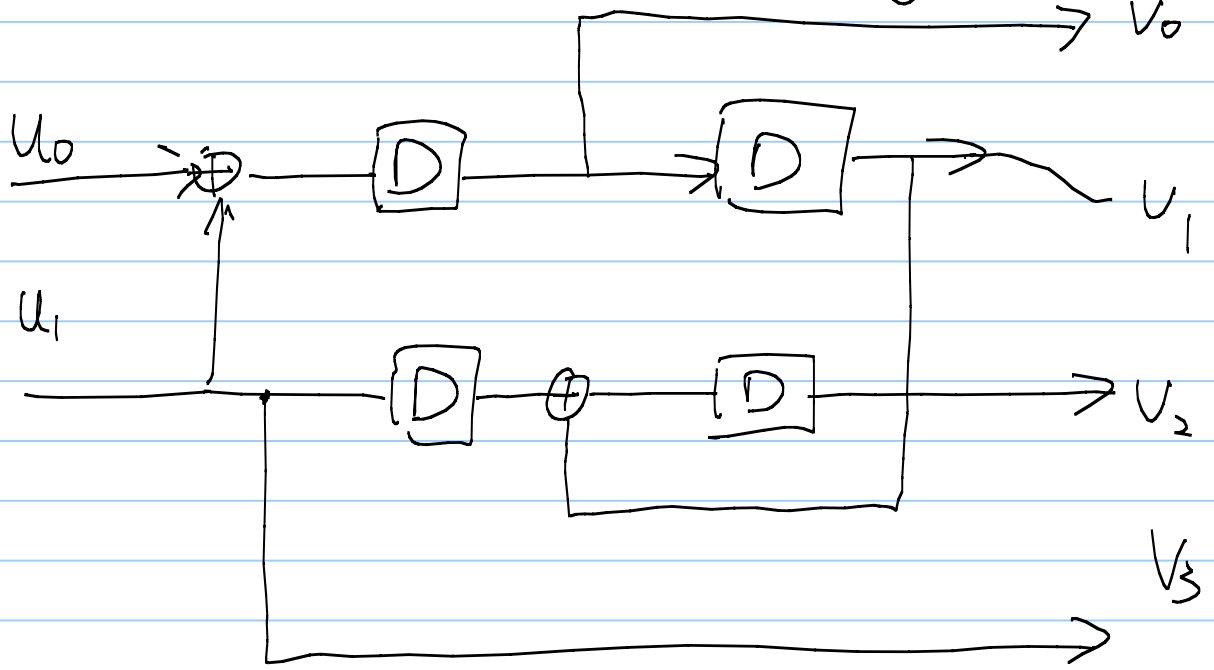
It is very cumbersome to perform optimal decoding, let alone evaluate its performance.

Usually can be done only for small codes

* Unless we focus on codes of special structure.

* Binary convolutional codes. by Elias 1955

A special class of binary linear codes



Features of convolutional codes

① Serial input (single bit or multiple bits.

$u_0\ u_1 \mid u_2\ u_3 \mid u_4\ u_5$ . . .

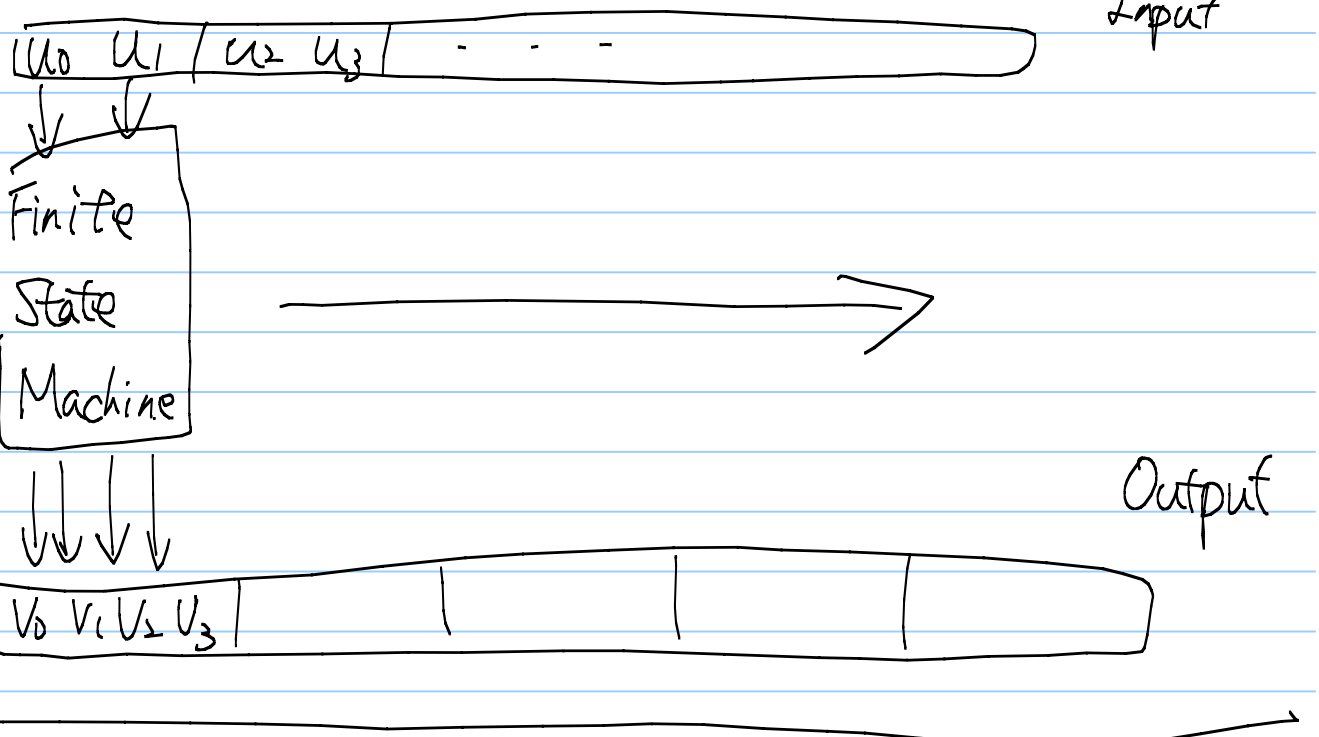
② Serial output

$v_0\ v_1\ v_2\ v_3 \mid v_4\ v_5\ v_6\ v_7$ . . .

③ Arbitrary connection among a finite # of delay units & binary ⊕ operations.

④ All memory or delay units are initialized to zero.

A more general form, sometimes called the trellis code is based on the finite state machine



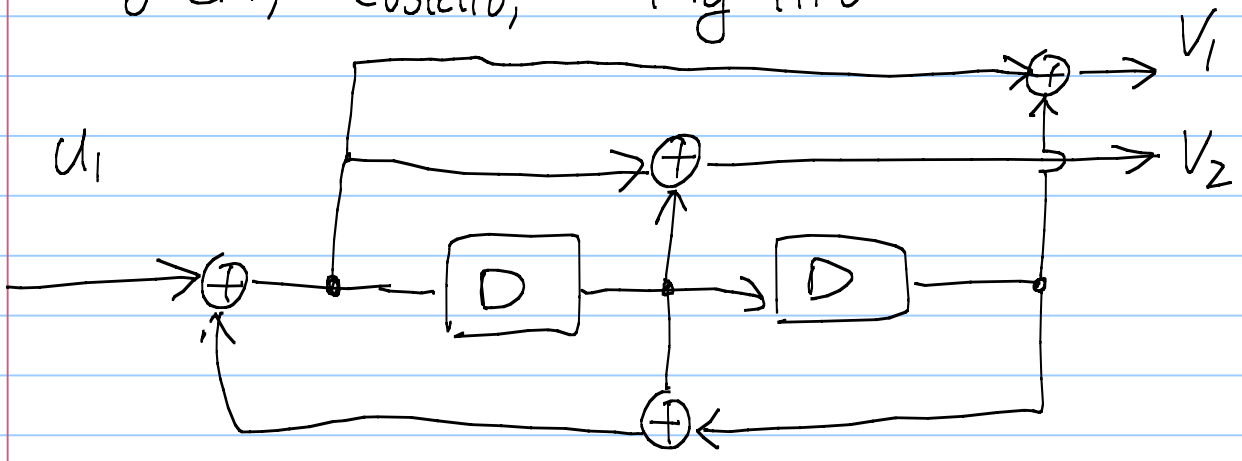
Properties of the convolutional code

① Still a binary linear code. (since the delay, ⊕ used for the FSM are linear.)

Since the input/output can be viewed as a discrete-time linear time-invariant system

$\Rightarrow$ P_X is uniform.

A running example Lin, Costello, Fig 11.6



② The code rate is $\frac{\#\text{ of } u \text{ bits}}{\#\text{ of } v \text{ bits}}$

$= \frac{1}{2}$ in our example

③ Encoding complexity: $O(n(\text{Complexity of FSM}))$

✱ ④ Complexity of "optimal decoding"

$O(|\#\text{ states}| \times n)$

We will discuss it later.

③+④ $\Rightarrow$ scalable for large n, Long codewords have better performance. (PRNG)

⑤ Early pseudo-random number generators are based on shift registers with feedback.

$\Rightarrow$ the $\vec{X}$ codeword distribution looks random.