WEEK 12

Error Correcting Codes

Linear Block Codes

A code word consists of n digits $c_1, c_2, ..., c_n$, and a data word consists of k digits $d_1, d_2, \ldots,$ dk, they are n- and k-dimensional vectors. We shall use row matrices to represent these words,

$$C = (c_1, c_2, \ldots, c_n),\ d = (d_1, d_2, \ldots, d_k)$$

For the general case of linear block codes, all the n digits of c are formed by linear combinations (modulo-2 additions) of k data digits. A special case where $c_1 = d_1, c_2 = d_2, ..., c_k = d_k$, and the remaining digits from c_{k+1} to c_n are linear combinations of $d_1, d_2, \ldots, d_k$ is known as a systematic code. Thus in a systematic code, the first k digits of a code word are the data digits and the last $m = n - k$ digits are the *parity-check digits*, formed by linear combinations of data digits $d_1, d_2, \ldots, d_k$

$$\begin{aligned}
c_1 &= d_1 \\
c_2 &= d_2 \\
&\vdots \\
c_k &= d_k \\
c_{k+1} &= h_{11}d_1 \oplus h_{12}d_2 \oplus \cdots \oplus h_{1k}d_k \\
c_{k+2} &= h_{21}d_1 \oplus h_{22}d_2 \oplus \cdots \oplus h_{2k}d_k \\
&\vdots \\
c_n &= h_{m1}d_1 \oplus h_{m2}d_2 \oplus \cdots \oplus h_{mk}d_k
\end{aligned}$$

or

$$\boldsymbol{c} = \boldsymbol{dG}$$

where

$$\boldsymbol{G} = \left[\underbrace{\begin{matrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ & & \vdots & & \\ 0 & 0 & 0 & \cdots & 1 \end{matrix}}_{\boldsymbol{I}_k (k \times k)} \quad \underbrace{\begin{matrix} h_{11} & h_{21} & \cdots & h_{m1} \\ h_{12} & h_{22} & \cdots & h_{m2} \\ & & \vdots & \\ h_{1k} & h_{2k} & \cdots & h_{mk} \end{matrix}}_{\boldsymbol{P} (k \times m)}\right]$$

⊕ this symbol means the modulo-2 addition, that's it:

0⊕0=1⊕1=0
0⊕1=1⊕0=1

The k x n matrix G is called the **generator matrix,** which can be partitioned into a k x k identity matrix and a k x m matrix P. All the elements of P are either 0 or 1. The code word can be expressed as

$$\begin{aligned}
c = dG &= d[I_k, P] \\
= [d, dP] &= [d, c_p]
\end{aligned}$$

So, according to this equation, Thus, knowing the data digits, we can calculate the check digits.

Example

For a (6, 3) code, the generator matrix G is

$$G = \left[\underbrace{\begin{matrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{matrix}}_{I_k} \quad \underbrace{\begin{matrix}1 & 0 & 1\\ 0 & 1 & 1\\ 1 & 1 & 0\end{matrix}}_{P}\right]$$

For all eight possible data words, find the corresponding code words, and verify that this code is a single-error correcting code.

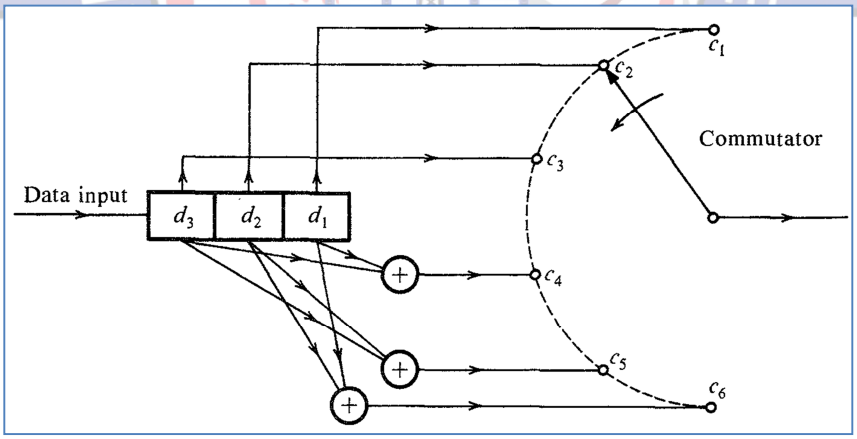
The eight possible data word are the binary form from 000 to 111

We shall find the code word for this data from the preceding procedure,

c= d*G, this yields the 8 code word shown in table

Data Word d	Code Word c
111	111000
110	110110
101	101011
100	100101
011	011101
010	010011
001	001110
000	000000

Note that the distance between any two code words is at least 3. Hence, the code can correct at least one error. Figure below shows a possible encoder for this code, using a three-digit shift register and three modulo-2 adders.



Linear Block Code Encoder

Hamming distance is the number of different bits between two codes

Code weight is the number of 1's in a certain code

1011101 code weight =5
1110100 code weight =4
& the hamming distance =3

LBC Decoding Procedure

Let us consider some code word properties that could be utilized for the purpose of decoding. From the fact that the modulo-2 sum of any sequence with itself is zero, we get

$$\boldsymbol{dP} \oplus \boldsymbol{c}_p = \underbrace{[\boldsymbol{d} \quad \boldsymbol{c}_p]}_{\boldsymbol{c}} \begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{I}_m \end{bmatrix} = 0$$

$$\boldsymbol{cH}^T = 0 \qquad \boldsymbol{H}^T = \begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{I}_m \end{bmatrix}$$

and the transpose: $\boldsymbol{H} = [\boldsymbol{P}^T \quad \boldsymbol{I}_m]$ is called the **parity-check matrix.**

Consider the received word r, Because of possible errors caused by channel noise; *r* in general differs from the transmitted code word c,

$$r = c \oplus e$$

where the error word (or error vector) *e*, is also a row vector of *n* elements. For example, if the data word 100 in the last example is transmitted as a code word 100101, and the channel noise causes a detection error in the third digit, then

$$c = 100101\,, r = 101101$$
$$and, e = 001000$$

Thus, an element 1 in $\boldsymbol{e}$ indicates an error in the corresponding position, and 0 indicates no error. The Hamming distance between $\boldsymbol{r}$ and $\boldsymbol{c}$ is simply the number of l's in $\boldsymbol{e}$.
Suppose the transmitted code word is $\boldsymbol{c_i}$ and the channel noise causes an error $\boldsymbol{e_i}$, making the received word $\boldsymbol{r = c_i + e_i}$ If there were no errors, that is, if e = 000000, then $\boldsymbol{rH^T = 0}$. But because of possible channel errors, $\boldsymbol{rH^T}$ is in general a non-zero row vector $\boldsymbol{s}$, called the **syndrome,**

$$\begin{aligned} \boldsymbol{s} &= \boldsymbol{r}\boldsymbol{H}^T \\ &= (\boldsymbol{c}_i \oplus \boldsymbol{e}_i)\boldsymbol{H}^T \\ &= \boldsymbol{c}_i H^T \oplus \boldsymbol{e}_i \boldsymbol{H}^T \\ &= \boldsymbol{e}_i \boldsymbol{H}^T \end{aligned}$$

Unfortunately, knowledge of $\boldsymbol{s}$ does not allow us to solve uniquely for $\boldsymbol{e_i}$. This is because $\boldsymbol{r}$ can also be expressed in terms of code words other than c,. Thus,

$$\boldsymbol{r = c_j \oplus e_j}$$

Hence,

$$\boldsymbol{s = (c_j \oplus e_j)H^T = e_j H^T}$$

Because there are $\boldsymbol{2^k}$ possible code words,

$$\boldsymbol{s = eH^T}$$

is satisfied by $\boldsymbol{2^k}$ error vectors. To give an example, if a data word $\boldsymbol{d = 100}$ is transmitted by a code word **100101** in the last example, and if a detection error is caused in the third digit, then the received word is **101101.** In this case we have $\boldsymbol{c = 100101}$ and $\boldsymbol{e = 001000.}$ But the same word could have been received if **c = 101011** and $\boldsymbol{e}$ **= 000110,** or if $\boldsymbol{c = 010011}$ and $\boldsymbol{e}$ =

111110, and so on. Thus, there are eight possible error vectors (2^k error vectors). Which vector shall we choose? For this, we must define our decision criterion. One reasonable criterion is the maximum-likelihood rule where, if we receive $\boldsymbol{r}$, then we decide in favor of that $\boldsymbol{c}$ for which $\boldsymbol{r}$ is most likely to be received. In other words, we decide " c_i transmitted " if

$$P(\boldsymbol{r}|\boldsymbol{c}_i) > P(\boldsymbol{r}|\boldsymbol{c}_k) \qquad \text{all } k \neq i$$

For a BSC, this rule gives a very simple answer. Suppose the Hamming distance between $\boldsymbol{r}$ and c_i is d, that is, the channel noise causes errors in d digits. Then if P_e is the digit error probability of a BSC,

$$P(\boldsymbol{r}|c_i) = P_e^d(1 - P_e)^{n-d} = (1 - P_e)^n \left(\frac{P_e}{1 - P_e}\right)^d$$

If $P_e < 0.5$, then; *P(r|ci)* is a monotonically decreasing function of d because $\left(\frac{P_e}{1-P_e}\right) < 1$. Hence, to maximize *P(r|ci)*, we must choose that ci which is closest to r; that is, we must choose the error vector with the smallest number of l's. A vector with the smallest number of 1's is called the **minimum-weight vector.**

Example:
A (6,3) code is generated according to the generating matrix given below. The receiver receives *r* = 100011. Determine the corresponding data word if the channel is a BSC and the maximum-likelihood decision is used.

$$\boldsymbol{G} = \begin{bmatrix} \underbrace{\begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix}}_{I_k} & \underbrace{\begin{matrix}1&0&1\\0&1&1\\1&1&0\end{matrix}}_{P} \end{bmatrix}$$

Sol:
We have:

$$s = rH^T$$

$$= [1\ 0\ 0\ 0\ 1\ 1] \begin{bmatrix} 1&0&1\\0&1&1\\1&1&0\\1&0&0\\0&1&0\\0&0&1 \end{bmatrix}$$

$$= [1\ 1\ 0]$$

Also we have:

$$\boldsymbol{c} = r \oplus \boldsymbol{e}$$

e should satisfy:

$s=[1\ 1\ 0] = eH^T$

$$=[e_1\ e_2\ e_3\ e_4\ e_5\ e_6]\begin{bmatrix}1&0&1\\0&1&1\\1&1&0\\1&0&0\\0&1&0\\0&0&1\end{bmatrix}$$

We see that e = 001000 satisfies this equation. But so does e = 000110, or 010101, or 011011, or 111110, or 110000, or 101101, or 100011. The suitable choice, the minimum- weight e, is 001000. Hence,

$$c = 100011 \oplus 001000 = 101011$$

The decoding procedure described recently is quite disorganized. A systematic procedure would be to consider all possible syndromes and for each syndrome associate a minimum-weight error vector. For instance, the single-error correcting code in the recent example has a syndrome with three digits. Hence, there are eight possible syndromes. We prepare a table of minimum-weight error vectors corresponding to each syndrome (see Table). This table can be prepared by considering all possible minimum-weight error vectors and computing $\boldsymbol{s}$ for each of them. The first minimum-weight error vector **000000** is a trivial case that has the syndrome **000.** Next, we consider all possible unit weight error vectors. There are six such vectors: **100000, 010000, 001000, 000100, 000010, 000001.** This still leaves one syndrome, **111,** that is not matched with some error vector. Since all unit-weight error vectors are exhausted, we must look for error vectors of weight 2.

$\boldsymbol{e}$	$\boldsymbol{s}$
000000	000
100000	101
010000	011
001000	110
000100	100
000010	010
000001	001
100010	111

We find that for the first seven syndromes, there is a unique minimum-weight vector $\boldsymbol{e}$. But for $\boldsymbol{s}$ = **111,** the error vector $\boldsymbol{e}$ has a minimum weight of 2, and it is not unique. For example, $\boldsymbol{e}$ = **100010** or **010100** or **001001** all have $\boldsymbol{s}$ = **111,** and all three e's are minimum weight (weight 2). In such a case, we can pick any one of these e's as a **correctable** error pattern. In the mentioned table, $\boldsymbol{e}$ = **100010** was picked as the double-error correctable pattern. This means the present code can correct all six single-error patterns and one double-error pattern **(100010).** For instance, if **c = 101011** is transmitted and the channel noise causes the double error **100010,** the received vector $\boldsymbol{r}$ = **001001,** and:

$$\boldsymbol{s} = \boldsymbol{r}\boldsymbol{H}^T = [1\,1\,1]$$

From this table, we see that corresponding to $\boldsymbol{s = 111}$ is $\boldsymbol{e = 100010}$, and we immediately decide $\boldsymbol{c = r \oplus e = 101011}$. Note, however, that this code will not correct double-error patterns other than **100010.** Thus, this code not only corrects all single errors but one double-error pattern as well (only when hamming over satisfied).

In general for any coding scheme, the minimum hamming distance d_{min}.

The code can detect $E_{det} = d_{min}-1$, and can correct $E_{correct} = (d_{min}- 1)/2$

Given that d_{min} is measured on the code word itself, see the table for all code words generated.

For our example, the all zero code word is ignored since it has no distance, the minimum number of 1's is 3 which represent d_{min} , this code can detect 2 errors & can fix single error only.

Here is some example for different correcting code with their efficiencies:

	n	k	Code	Code Efficiency (or Code Rate)
Single-error correcting, $t = 1$	3	1	(3, 1)	0.33
	4	1	(4, 1)	0.25
Minimum code separation 3	5	2	(5, 2)	0.4
	6	3	(6, 3)	0.5
	7	4	(7, 4)	0.57
	15	11	(15, 11)	0.73
	31	26	(31, 26)	0.838
Double-error correcting, $t = 2$	10	4	(10, 4)	0.4
	15	8	(15, 8)	0.533
Minimum code separation 5				
Triple-error correcting, $t = 3$	10	2	(10, 2)	0.2
	15	5	(15, 5)	0.33
Minimum code separation 7	23	12	(23, 12)	0.52

In case $\boldsymbol{n}$ and $\boldsymbol{k}$ were to satisfy the hamming bound exactly, we would have only single-error correction ability. This is the case for the (7,4) code, which can correct all single-error patterns only.
Thus for systematic decoding, we prepare a table of all correctable error patterns and the corresponding syndromes. For decoding, we need only calculate $\boldsymbol{s = rH^T}$ and, from the decoding table, find the corresponding $\boldsymbol{e}$. The decision is $\boldsymbol{c = r \oplus e}$.

It is still not clear how to choose coefficients of the generator or parity-check matrix. Unfortunately, there is no systematic way to do this, except for the case of single-error correcting codes, also known as *Hamming codes.* Let us consider a single-error correcting (7, 4) code. This code satisfies the Hamming bound exactly, and we shall see that a proper code can be constructed. In this case $m = 3$, and there are seven nonzero syndromes, and because n

= 7, there are exactly seven single-error patterns. Hence, we can correct all single-error patterns and no more. Consider the single-error pattern e = 1000000. Because

$$s = eH^T$$

eH^T will be simply the first row of H^T. Similarly, for e = 0100000, s = eH^T will be the second row of H^T, and so on. Now for unique decodability, we require that all seven syndromes corresponding to the seven single-error patterns be distinct. Conversely, if all the seven syndromes are distinct, we can decode all the single-error patterns. This means that the only requirement on H^T is that all seven of its rows be distinct and nonzero. Note that H^T is an $(n$ x $n - k)$ matrix (i.e., 7 x 3 in this case). Because there exist seven nonzero patterns of three digits, it is possible to find seven nonzero rows of three digits each. There are many ways in which these rows can be ordered. But we must remember that the three bottom rows must form identity matrix I_m

One possible form of H^T is:

$$\boldsymbol{H}^T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{I}_m \end{bmatrix}$$

$$\boldsymbol{G} = [\boldsymbol{I}_k \quad \boldsymbol{P}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Thus when d = 1011 is transmitted, the corresponding code word c = 1011001, and so forth.

A general (n, k) code has m-dimensional syndrome vectors $(m = n - k)$. Hence, there are $2^m - 1$ distinct nonzero syndrome vectors that can correct $2^m - 1$ single-error patterns. Because in an (n, k) code there are exactly n single-error patterns, all these patterns can be corrected if:

$$2^m \geq 1 + n$$

Example: Determine the parity check matrix $\boldsymbol{H}$ for the linear block code (7, 4) generated by the generator matrix:

$$\boldsymbol{G} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Since

$$\boldsymbol{G} = \underbrace{\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}}_{\text{submatrix } \boldsymbol{P}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{submatrix } \boldsymbol{I}}$$

the parity check matrix $\boldsymbol{H}$ is constructed using these submatrices:

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

What about H^{T} ? Find it.