ECEN 604 - Channel Coding for Communication
Lectures by Graduate Students

Narayanan Rengaswamy
Department of Electrical and Computer Engineering
Texas A&M University, College Station, TX 77843

October 2015

Lecture 3

1 Linear Block Codes

In this discussion, we will define linear block codes over a group and, subsequently, a finite field and study its properties. We will look at an elegant way of constructing and storing a code, via its generator and parity-check matrices, and its encoding and decoding methods. When we discuss the decoding strategy, we will leverage the concept of coset tables efficiently. Throughout this discussion, we will use Hamming codes as our test bed to understand the concepts. Refer to [1, Chapter 3] for a thorough discussion of this material.

Consider a message source emitting message symbols from an alphabet $\mathcal{X}$ with q symbols that forms an additive group. We will see at the end of Section 1.4.1 that even though linear codes can be defined over an additive group, we need the alphabet to be a field to implement these codes. We refer to a vector $\underline{m} = (m_0, m_1, \ldots, m_{k-1}) \in \mathcal{X}^k$ as a k-tuple, where $\mathcal{X}^k = \mathcal{X} \times \mathcal{X} \times \cdots \times \mathcal{X}$ is the product set of the alphabet.

In the most general case, a block code is a set of codewords. For example, let us take the example of the $(6, 3)$ binary block code $\mathcal{C}$ that we have discussed earlier. So, we have the alphabet as $\mathcal{X} = \{0, 1\}$, which implies that $\underline{m} \in \mathcal{X}^3$ and $\underline{c} \in \mathcal{X}^6$. Since the inputs for this code are $k = 3$-bit messages, the code contains $M = 2^k = 8$ codewords, each of length $n = 6$. The messages and codewords for this code are given in Table 1.

This code is said to be a **block code** of length $n = 6$, dimension $k = 3$ and size $M = 8$. Also, it can be verified that the mod-2 sum of any two codewords from this code also lies in the code. This can be formally expressed as below.

$$\underline{x} + \underline{y} \in \mathcal{C} \ \forall \ \underline{x}, \underline{y} \in \mathcal{C}.$$

Hence, this is a **linear** block code.

$\underline{m}$	$\underline{c}$	$\underline{m}$	$\underline{c}$
000	000000	100	101100
001	110001	101	011101
010	011010	110	111110
011	101011	111	000111

Table 1: The messages and codewords of the 1-error correcting $(6, 3)$ linear block code.

In this case, since we have a binary alphabet, a linear combination of multiple codewords just involves the inclusion or exclusion of each of them. So, we only need (point-by-point) vector addition defined on the alphabet $\mathcal{X}^k$; there is no need for a scalar multiplication operation defined on this alphabet. Hence, it suffices if $\mathcal{X}$ is a group under mod-2 addition. But, to implement efficient encoders and decoders for the code, instead of having lookup tables, we will have to perform matrix-vector and vector-vector products as we will see shortly. Therefore, the alphabet also needs to be an Abelian group under a multiplication operation defined on its elements. Additionally, we might also have a larger-sized alphabet which will give rise to non-trivial scalar multiplication on vectors.

Hence, we need the alphabet to be a **field** under appropriate addition and multiplication operations. Once the alphabet is a field, the code $\mathcal{C} \subseteq \mathcal{X}^n$ is a **vector subspace**. For example, consider the same $(6, 3)$ code, but, defined on a ternary alphabet $\mathcal{X} = \{0, 1, 2\}$, which is a field under mod-3 addition and multiplication operations. Then, there can be $3^k = 27$ 3-ary messages (instead of binary) and we need an equal number of codewords. In this case, it is possible to construct a vector $(2 \cdot \underline{x} + \underline{y})$ given $\underline{x}, \underline{y} \in \mathcal{C}$. For the code to still be linear, such a combination should also lead to another codeword in the same code. Now, let us generalize and formalize these concepts.

Definition 1. An (n, k) **block code** $\mathcal{C}$ over a q**-ary** alphabet $\mathcal{X}$ is a set of q^k n-tuples called **codewords** or **code vectors**. Associated with the code is an **encoder** that maps a k-tuple message $\boldsymbol{m} \in \mathcal{X}^k$ to its associated codeword.

Typically, the mapping between messages and codewords is bijective so that error correction is easier. But, there can be multiple ways of mapping messages to codewords.

Definition 2. A block code $\mathcal{C}$ over a **field** $\mathbb{F}_q$, of q symbols, with q^k codewords of length n is a q-ary **linear** (n, k) code if and only if the q^k codewords of the code form a k**-dimensional vector subspace** of the vector space of all q-ary n-tuples $\mathbb{F}_q^n$.

Remark: All linear codes contain the all-zero vector, i.e. the all-zero codeword.

Definition 3. For an (n, k) linear block code $\mathcal{C}$, n is called its **length** and k is called its **dimension**.

Definition 4. The **size** of an (n, k) linear block code $\mathcal{C}$ is $M = |\mathcal{C}| = |\mathcal{X}|^k = q^k$.

Definition 5. The **rate** of an (n, k) linear block code $\mathcal{C}$ is $R = \dfrac{k}{n}$.

Remark: A code can have more than $|\mathcal{X}|^k$ codewords and in that case, $M \geq |\mathcal{X}|^k$ and the rate of the code is defined more generally as $R = \dfrac{\log_{|\mathcal{X}|} M}{n}$.

Definition 6. The **Hamming weight w_H** of a vector $\underline{x} = (x_1, \ldots, x_n) \equiv x_1^n$ is defined as the number of non-zero entries in it.

$$w_H(\underline{x}) \triangleq \sum_{i=1}^{n} \mathbb{I}(x_i \neq 0),$$

where $\mathbb{I}(\cdot)$ is the indicator function which returns 1 if its argument is true.

Definition 7. The **minimum weight $w_{\min}$** of a code $\mathcal{C}$ is the weight of that codeword $\underline{c} \in \mathcal{C}$ with least Hamming weight among all the codewords in $\mathcal{C}$.

$$w_{\min}(\mathcal{C}) \triangleq \min_{\underline{c} \in \mathcal{C}} w_H(\underline{c}).$$

Definition 8. The **Hamming distance d_H** between any two vectors x_1^n and y_1^n is defined as the number of indices in which they differ.

$$d_H(\underline{x}, \underline{y}) \triangleq \sum_{i=1}^{n} \mathbb{I}(x_i \neq y_i) = w_H(\underline{x} - \underline{y}).$$

Definition 9. The **minimum distance $d_{\min}$** of a code $\mathcal{C}$ is the smallest Hamming distance between any pair of codewords in $\mathcal{C}$.

$$d_{\min}(\mathcal{C}) \triangleq \min_{\substack{\underline{c_1}, \underline{c_2} \in \mathcal{C} \\ \underline{c_1} \neq \underline{c_2}}} d_H(\underline{c_1}, \underline{c_2}) = \min_{\substack{\underline{c_1}, \underline{c_2} \in \mathcal{C} \\ \underline{c_1} \neq \underline{c_2}}} w_H(\underline{c_1} - \underline{c_2}).$$

Remark: An (n, k) linear block code of minimum distance $d_{\min}$ is sometimes represented as an $(n, k, d_{\min})$ code.

Theorem 10. *For a linear block code $\mathcal{C}$, the minimum distance is equal to the minimum weight of the code,*

$$\textit{i.e.} \quad d_{\min}(\mathcal{C}) = w_{\min}(\mathcal{C}).$$

Proof. Let $\underline{c_1}, \underline{c_2} \in \mathcal{C}$ be at Hamming distance d from each other and $(\underline{c_1} - \underline{c_2}) \in \mathcal{C}$ have Hamming weight w. Then, since the code is linear, the following holds true.

$$d = d_H(\underline{c_1}, \underline{c_2}) = d_H(\underline{c_1} - \underline{c_2}, \underline{0}) = w_H(\underline{c_1} - \underline{c_2}) = w \geq w_{\min}.$$

Let $w = w_{\min}$, i.e. $(\underline{c_1} - \underline{c_2})$ is a minimum weight codeword in $\mathcal{C}$. If $d > d_{\min}$, then that means there exists two other codewords at a distance $d_{\min} < d$ from each other, i.e. whose difference is a codeword of weight less than $w_{\min}$, which is impossible. Hence, $d_{\min} = w_{\min}$. $\square$

1.1 The Generator Matrix Representation

Since a linear block code $\mathcal{C}$ is a k-dimensional vector subspace of $\mathbb{F}_q^n$, the code can be represented by its basis vectors $\underline{\boldsymbol{g}}_0, \underline{\boldsymbol{g}}_1, \ldots, \underline{\boldsymbol{g}}_{k-1}$. This enables us to write any codeword $\underline{c} \in \mathcal{C}$ as a linear combination of these vectors so that

$$\underline{c} = m_0 \cdot \underline{\boldsymbol{g}}_0 + m_1 \cdot \underline{\boldsymbol{g}}_1 + \cdots + m_{k-1} \cdot \underline{\boldsymbol{g}}_{k-1}, \tag{1}$$

where $m_i \in \mathbb{F}_q$. The standard way of representing a linear block code is by stacking up these basis vectors to form a $k \times n$ **generator matrix** for the code as

$$G = \begin{bmatrix} \underline{g}_0 \\ \underline{g}_1 \\ \vdots \\ \underline{g}_{k-1} \end{bmatrix}_{k \times n}. \tag{2}$$

The **row space of the generator matrix** G is the linear block code $\mathcal{C}$. Hence, given a message vector $\underline{m} = [m_0 \; m_1 \; \cdots \; m_{k-1}]$, the codeword is obtained by

$$\underline{\boldsymbol{c}} = \underline{\boldsymbol{m}}\, \boldsymbol{G}. \tag{3}$$

There is no unique generator matrix for a given code $\mathcal{C}$. Given a generator matrix G, it is possible to perform row operations on the matrix to arrive at another generator matrix G' for the same code $\mathcal{C}$ which will retain the same q^k codewords but will only change the mapping between messages and codewords.

Example

Consider the code $\mathcal{C} = \{111, 100, 010, 001\}$, which is an odd parity code. This is not a linear code since $111 \oplus 100 = 011 \notin \mathcal{C}$. But the even parity code $\mathcal{T} = \{000, 110, 011, 101\} = \{0,1\}^3 \setminus \mathcal{C}$ is linear. Notice that while the odd parity code does not contain the all-zero vector, the even parity code does contain it. The presence of this vector is a necessary condition for the code to be linear.

The code $\mathcal{T}$ is a subgroup of $\{0,1\}^3$. Hence, $\mathcal{T} \oplus (001) = \{001, 111, 010, 100\}$ is a coset of $\mathcal{T}$. It is easy to see that this is the odd parity code $\mathcal{C}$. We have constructed the coset $\mathcal{C}$ using the subgroup $\mathcal{T}$ of the group $\{0,1\}^3$.

Since a generator matrix describes the code completely, it is not required to store the complete code book. This is a huge savings in memory space because, while the code size is q^k vectors of length n, we only need to store k vectors of length n!

Definition 11. Let $\mathcal{C}$ be an (n, k) block code (not necessarily linear). An encoder for this code is **systematic** if the message symbols $m_0, m_1, \ldots, m_{k-1}$ may be found explicitly and unchanged in the codeword, i.e. k of the n codeword symbols are the message symbols itself and the rest are parity symbols.

A generator for such an encoder is a **systematic generator** and can be expressed as

$$\boldsymbol{G_{\text{sys}} = [P\ I_k]}, \tag{4}$$

where P is the $(n-k)\times(n-k)$ matrix that generates the parities and I_k is the $k\times k$ identity matrix that retains the message symbols as is.

Definition 12. Two linear codes which are equal under a column permutation, i.e. reordering of code symbols, are said to be **equivalent**. The generators of the two codes vary only by a column permutation.

Example

Consider the $(7, 4)$ binary Hamming code. This is a 4-dimensional vector subspace of $\mathbb{F}_2^7 = \{0, 1\}^7$ and consists of $2^4 = 16$ binary codewords of length 7. It can be verified that this code has a minimum distance $d_{\min} = 3$. The code has a generator matrix

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

and a systematic generator matrix

$$G_{\text{sys}} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Hence, if the message vector is $\underline{m} = [m_0\ m_1\ m_2\ m_3]$, then the three parity symbols c_0, c_1, c_2 of the *systematic* codeword $\underline{c} = [c_0\ c_1\ c_2\ m_0\ m_1\ m_2\ m_3]$ are generated as

$$\begin{aligned} c_0 &= m_0 + m_2 + m_3, \\ c_1 &= m_0 + m_1 + m_2, \\ c_2 &= m_1 + m_2 + m_3. \end{aligned}$$

1.2 Dual Codes and The Parity-Check Matrix

We know from the theory of linear algebra that, if $\mathcal{V}$ is a subspace of a n-dimensional vector space, then there exists a **dual space** for $\mathcal{V}$, represented by $\mathcal{V}^\perp$, such that

$$dim(\mathcal{V}) + dim(\mathcal{V}^\perp) = n.$$

Definition 13. Give a linear block code $\mathcal{C}$ that is a k-dimensional subspace of $\mathbb{F}_q^n$, there exists a **dual code** for $\mathcal{C}$, represented as

$$\mathcal{C}^\perp = \{\underline{x} \in \mathbb{F}_q^n \mid \underline{x} \cdot \underline{y} = 0\ \forall\ \underline{y} \in \mathcal{C}\}, \tag{5}$$

which is an $(n-k)$-dimensional subspace of $\mathbb{F}_q^n$. $\underline{x} \cdot \underline{y}$ denotes the inner product of $\underline{x}$ and $\underline{y}$.

The basis for $\mathcal{C}^\perp$ is the set $\{\underline{\boldsymbol{h}}_{\boldsymbol{0}}, \underline{\boldsymbol{h}}_{\boldsymbol{1}}, \ldots, \underline{\boldsymbol{h}}_{\boldsymbol{n-k-1}}\}$ and the generator matrix is constructed as

$$H = G_{\mathcal{C}^\perp} = \begin{bmatrix} \underline{\boldsymbol{h}}_{\boldsymbol{0}} \\ \underline{\boldsymbol{h}}_{\boldsymbol{1}} \\ \vdots \\ \underline{\boldsymbol{h}}_{\boldsymbol{n-k-1}} \end{bmatrix}_{(n-k)\times n} . \tag{6}$$

This matrix H is called the **parity-check matrix** of the code $\mathcal{C}$.

Since the two codes are duals of each other, the following condition holds true for a given code $\mathcal{C}$:

$$\boldsymbol{G}\ \boldsymbol{H}^{\boldsymbol{T}} = \boldsymbol{O}, \tag{7}$$

where O is the $k \times (n-k)$ matrix of all zero entries. Hence, the necessary and sufficient condition for any length-n vector with q-ary elements to be a codeword of $\mathcal{C}$ is

$$\underline{\boldsymbol{c}}\ \boldsymbol{H}^{\boldsymbol{T}} = \underline{\boldsymbol{0}} \ \text{ or } \ \boldsymbol{H}\ \underline{\boldsymbol{c}}^{\boldsymbol{T}} = \underline{\boldsymbol{0}}^{\boldsymbol{T}}, \tag{8}$$

where $\underline{0}$ is the length $(n-k)$ row vector of all zeroes. It is easy to see that these are exactly the parity-check equations for the code $\mathcal{C}$.

A **systematic parity-check matrix** for a code $\mathcal{C}$ is given by

$$\boldsymbol{H}_{\mathbf{sys}} = [\boldsymbol{I}_{\boldsymbol{n-k}}\ -\boldsymbol{P}^{\boldsymbol{T}}], \tag{9}$$

where I_{n-k} is the size $(n-k)$ identity matrix and P is the matrix generating the parity symbols in the systematic generator for the code. Once again, it is important to note that there is no unique parity-check matrix for a given code.

Theorem 14. *The minimum weight, $w_{\min}$, of a code $\mathcal{C}$ is equal to the minimum number of dependent columns in its parity-check matrix H.*

Proof. We know that $H\ \underline{c}^T = \underline{0}^T\ \forall\ \underline{c} \in \mathcal{C}$. Let $w_H(\underline{c}) = w_{\min}$ and the non-zero indices of $\underline{c}$ be $\{c_{i_1}, c_{i_2}, \ldots, c_{i_w}\}$. Denote the columns vectors of H as $\underline{h}_0^T, \underline{h}_1^T, \ldots, \underline{h}_{n-1}^T$ (these are not the basis vectors of $\mathcal{C}^\perp$ used to construct H in its definition above). Then, we have

$$H\ \underline{c}^T = \underline{0}^T \Rightarrow c_{i_1}\ \underline{h}_{i_1}^T + c_{i_2}\ \underline{h}_{i_2}^T + \cdots + c_{i_w}\ \underline{h}_{i_w}^T = 0.$$

This means that the columns $\{\underline{h}_{i_1}^T, \underline{h}_{i_2}^T, \ldots, \underline{h}_{i_w}^T\}$ are linearly dependent. But, since $\underline{c}$ is a vector of minimum weight in $\mathcal{C}$, there cannot be another vector, say $\underline{c}'$, of weight less than $w_H(\underline{c})$ that satisfies $H\ \underline{c'}^T = \underline{0}^T$. In other words, for $w' < w$, there are no w' columns of H that are linearly dependent. □

Example

For the $(7, 4)$ binary Hamming code with the systematic generator matrix, a systematic parity-check matrix is given by

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

It can be verified that the parity-check equations for a codeword $\underline{c} = [c_0\ c_1\ \cdots\ c_6] = [c_0\ c_1\ c_2\ m_0\ m_1\ m_2\ m_3]$ are

$$\begin{aligned} c_0 + c_3 + c_5 + c_6 &= 0, \\ c_1 + c_3 + c_4 + c_5 &= 0, \\ c_2 + c_4 + c_5 + c_6 &= 0. \end{aligned}$$

In the above parity-check matrix, observe that all non-zero length-3 binary vectors are present as columns in H. We will see in Section 1.4.2 that this defines the largest single error correcting code of length 7. In general, an **(n, k) binary Hamming code** is defined by

$$\boxed{n = 2^r - 1\ ;\ k = 2^r - 1 - r}$$

where r parity check bits are added to each message and r is the number of rows in the parity-check matrix H. The columns of the H matrix of the code are all the binary non-zero vectors of length r. This definition can be generalized to non-binary Hamming codes too.

Definition 15. For an (n, k) linear block code, $r = n - \log_q M$ is called its **redundancy** and is essentially equal to the number of parity check bits in every codeword of the code.

Theorem 16. *The minimum distance of all binary Hamming codes is $d_{\min} = 3$.*

Proof. By construction, the columns of the H matrix of the code are all the binary non-zero vectors of length r. So, the sum of any two columns of H will be another column in H. Hence, the minimum number of dependent columns in H is 3. □

1.3 Simple Bounds on Codes

Theorem 17. *(The **Singleton Bound**) The minimum distance of an (n, k) linear block code $\mathcal{C}$ with $|\mathcal{X}^k|$ codewords is upper bounded as*

$$d_{\min}(\mathcal{C}) \leq n - k + 1.$$

Proof. An (n, k) linear code has a parity-check matrix H whose $(n - k)$ rows are linearly independent, since they form the basis for the dual code $\mathcal{C}^{\perp}$. Since the row rank of a matrix is equal to its column rank, $\text{rank}(H) = n - k$. Hence, by the definition of the rank of a matrix, any set of $(n - k + 1)$ columns of H must be linearly dependent. □

Definition 18. An (n, k) code with $d_{\min} = n - k + 1$ is called a **maximum distance separable (MDS)** code.

The famous example for MDS codes are Reed-Solomon (RS) codes, which we will study later in this course.

The vectors at a (Hamming) distance at most t from a codeword form a "sphere" called the **Hamming sphere** of radius t.

Definition 19. The **volume** of a Hamming sphere of dimension m over $|\mathcal{X}| = q$ and radius t is given by

$$V_q(n,t) \triangleq \sum_{j=0}^{t} \binom{n}{j} (q-1)^j.$$

Theorem 20. *(The **Hamming Bound**) A t-random error correcting q-ary (n,k) code $\mathcal{C}$ with M codewords must satisfy*

$$M \leq \frac{q^n}{V_q(n,t)}.$$

Proof. If a code corrects all t-error patterns, then all the radius t Hamming spheres must be disjoint. Hence, since the total volume of the n-dimensional vector space is q^n, we have

$$q^n \geq M \; V_q(n,t).$$

Equivalently, the redundancy r of the code must satisfy

$$r \geq \log_q V_q(n,t) \qquad \square$$

Definition 21. A code satisfying the Hamming bound with equality is said to be a **perfect code**.

Example

Consider the $(7,4)$ binary Hamming code. We have $V_2(7,1) = 8$ and therefore,

$$\frac{2^7}{V_2(7,1)} = \frac{128}{8} = 16 = M.$$

In general, *all Hamming codes are perfect codes.*

Theorem 22. *(The **Gilbert-Varshamov Bound**) There exists a length-n q-ary linear block code with M codewords and $d_{\min} \geq d$ if*

$$(M-1)V_q(n,d-1) \leq q^n - 1.$$

Proof. Consider the space of all q-ary n-tuples. The volume of the space is q^n. We know that for good error correction, the Hamming spheres of the target code should all be disjoint. From the space of all n-tuples, pick one vector as a codeword. This induces the removal of $V_q(n,d-1)$ codewords from the set of possible codewords. Then, pick another codeword from the remaining $q^n - V_q(n,d-1)$ n-tuples. After i such steps, we have at least $q^n - iV_q(n,d-1)$ vectors left. Hence, once we pick $(M-1)$ codewords, we must have at least one vector left to be chosen as a codeword.

$$\Rightarrow q^n - (M-1)V_q(n,d-1) \geq 1.$$

This condition implies that there exists a code with parameters n and M with $d_{\min} \geq d$. $\square$

1.4 Decoding

Let $\mathcal{C}$ be an (n, k) linear block code over a field $\mathbb{F}_q$ and $\underline{c} \in \mathbb{F}_q^n$ be the transmitted codeword. Then, the received word is represented as

$$\underline{r} = \underline{c} + \underline{e}, \tag{10}$$

where $\underline{e} \in \mathbb{F}_q^n$ is the error vector used to model the errors introduced by the channel.

Given a received vector $\underline{r}$, the decoder is some function $g : \mathbb{F}_q^n \to \mathcal{C} \cup \phi$ of it such that

$$g(\underline{r}) = \hat{\underline{c}}, \tag{11}$$

where $\hat{\underline{c}}$ is the estimated codeword and ϕ represents a decoder failure which is declared when the number of estimated errors in the received word are larger than the error correcting capability of the decoder.

Definition 23. (The **MAP Decoder**) The optimal decoder that maximizes $p(\underline{c} \mid \underline{r})$ is called the *maximum-a-posteriori decoder*.

$$g_{\text{MAP}}(\underline{r}) \triangleq \arg\max_{\underline{c} \in \mathcal{C}} p(\underline{c} \mid \underline{r}).$$

Definition 24. (The **ML Decoder**) In a code with uniform distribution on the codewords, the optimal decoder that maximizes $p(\underline{c} \mid \underline{r})$ is the one that maximizes $p(\underline{r} \mid \underline{c})$ and is called the *maximum likelihood decoder*.

$$g_{\text{ML}}(\underline{r}) \triangleq \arg\max_{\underline{c} \in \mathcal{C}} p(\underline{r} \mid \underline{c}).$$

1.4.1 Maximum Likelihood (ML) Decoding

For an (n, k) linear block code $\mathcal{C}$ over a group $\mathbb{F}_q$, we use the concept of coset tables in groups to design a decoding algorithm. Recollect that the set of all n-tuples with q-ary coordinates forms an Abelian group $\mathbb{F}_q^n$ under vector addition. The code $\mathcal{C}$ is a normal subgroup of $\mathbb{F}_q^n$. So, we construct the coset table of this subgroup by choosing the coset leaders as the vectors of minimum weight that have not already appeared in the table. The resulting table is called a **standard array**. This table consists of all possible received vectors. Hence, once we locate the received vector in the table, the decoded codeword is the codeword present in the first row of the column containing the received vector. The coset leaders are called the set of all **correctable error patterns**. This standard array decoder is the ML decoder for the BSC channel.

Example

Consider the $(5, 2)$ binary code defined by

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad G = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

We have $M = 4$ codewords and the code is a subgroup of $\{0,1\}^5$ listed by its codewords as

$$\mathcal{C} = \{00000, 10110, 11101, 01011\}.$$

Standard Array

00000	10110	11101	01011
10000	00110	01101	11011
01000	11110	10101	00011
00100	10010	11001	01111
00010	10100	11111	01001
00001	10111	11100	01010
11000	01110	00101	10011
01100	11010	10001	00111

The first row is the set of all codewords in the code $\mathcal{C}$, i.e. the subgroup of $\mathbb{F}_2^5$. The vectors in the first column are the coset leaders and they form the set of error patterns that can be corrected by the code. Every row is formed by choosing the coset leader as a minimum weight vector that has not already appeared in the table. Considering this as an error pattern, the rest of the elements in the row are formed by adding this error vector to the corresponding codeword in the columns. Finally, the table contains all possible binary vectors of length 5.

Given a received vector $\underline{r}$, locate it in this table; say the location is row i and column j. Since the corresponding coset leader of row i is the most probable error pattern that affected the codeword during transmission through the channel, $\underline{r}$ is decoded to the codeword in the first row of column j. Since all single error patterns are coset leaders in the standard array, this code has an error correcting capability of $t = 1$. But, since only two double error patterns are present as coset leaders, it is better to declare decoding failures if the received vector lies in the last two rows. This is because the set of received vectors for all 2-error patterns are not distinct. Also, if a transmitted codeword suffers multiple errors to be received as the vector $\underline{r}$, then this leads to wrong decoding since the code can correct only single error patterns. Hence, the ML decoder can, potentially, *miscorrect* errors.

Since this code is small in size, it is feasible to construct the standard array and decode. However, for larger codes the memory and computational requirements are humongous. Therefore, we resort to suboptimal decoding strategies in general. One such strategy is the bounded distance decoder.

Definition 25. A t-error correcting *bounded distance decoder* selects the codeword $\underline{c}$ given the received vector $\underline{r}$ if $d_H(\underline{r}, \underline{c}) \leq t$. If no such codeword exists, then a decoder failure is declared.

Recollect that for each codeword in the code, a Hamming sphere can be constrcuted with the codeword as its center and all non-codewords at a distance at most t from the center as other points inside the sphere. Then, given the received vector, the bounded distance decoder attempts to identify the Hamming sphere containing it and if it is successful, it decodes the received word to the center of that sphere. If none of the M spheres of the code contains that received vector, the decoder declares a failure so that $g(\underline{r}) = \phi$.

Theorem 26. *The error correcting capability of an (n, k) linear block code of minimum distance $d_{\min}$ is*

$$t = \lfloor \frac{d_{\min} - 1}{2} \rfloor.$$

Proof. The centers of any two Hamming spheres of the code, with radius t, are at a distance at least $d_{\min}$ from each other. Consider the closest case, i.e. when the two centers are exactly $d_{\min}$ away from each other. If all n-tuples in the space are to be classified into a unique sphere, no two spheres should even touch each other. Therefore, $d_{\min} \geq 2t + 1$. □

Remark: For a perfect code, the bounded distance decoder is the ML decoder since the Hamming spheres encompass all the n-tuples in the vector space.

Alphabet: Group or Field?

Note that if we perform encoding of messages using a lookup table for the codewords and perform decoding using the standard array, it suffices that the alphabet for the code be a group under addition and not necessarily a field. However, if we intend to use a generator matrix approach to store the code and encode, and use the parity-check matrix to decode (as we will show next), then we need to perform multiplication of elements from the alphabet which requires it to be a field.

1.4.2 Syndrome Decoding

The syndrome decoder is a space-efficient implementation of the ML decoder described above.

Definition 27. For an (n, k) linear block code $\mathcal{C}$ over a field $\mathbb{F}_q$ with parity-check matrix H, the $(n - k)$ **syndromes** for a received vector $\underline{r}$ are defined as

$$\underline{s} = \underline{r}\ H^T.$$

There are q^{n-k} possible syndromes. Clearly, $\underline{r} = \underline{c} + \underline{e}$ is a codeword if and only if $\underline{s} = \underline{0}$, since $\underline{c}\ H^T = \underline{0}\ \forall\ \underline{c} \in \mathcal{C}$. Hence, we have

$$\underline{s}^T = H\ \underline{r}^T = H\ \underline{e}^T. \tag{12}$$

We see that the syndromes are linear combinations of columns of H. Therefore, starting from the smallest weight error patterns, we can make a table for mapping all possible syndromes to their corresponding, unique, error patterns. These form the correctable error patterns for

the decoder. Now, instead of storing the standard array, we only need to store the set of syndromes and their corresponding error patterns in a **syndrome table**. Once the error pattern, $\underline{\hat{e}}$, is estimated based on the syndrome table, the codeword is recovered as $\underline{\hat{c}} = \underline{r} - \underline{\hat{e}}$.

Even though this saves a lot of space, this still requires a lot of memory for long codes. Hence, there should be more algebraic structure in the code to simplify decoding, in terms of memory requirements.

Example

Once again, consider the $(7, 4)$ binary Hamming code. The parity-check matrix H consists of all the 7 binary non-zero vectors of length 3 as its columns. So, if we perform $\underline{s} = H\ \underline{e}^T$ for all the possible 7 single error patterns, we get the set of all unique non-zero syndromes. Hence, this is the largest code of length 7 that can correct all single bit errors, as mentioned earlier.

1.5 Performance of Linear Codes

Let $\mathcal{C}$ be an (n, k) linear code of minimum distance $d_{\min}$. Consider the bounded distance decoder for this code. Let us determine the probability of decoding error P_e on BSC(p), under this decoding strategy. The decoder corrects all patterns of t or less errors, where $t = \lfloor \frac{d_{\min}-1}{2} \rfloor$. Hence, the error probability can be upper bounded as

$$P_e \leq \sum_{i=t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i}. \tag{13}$$

Example

Consider a $t = 3$-random error correcting code on BSC(0.01) and let the requirement for the application be $P_e = 10^{-6}$. We need to find the blocklength n required to guarantee this performance. Since p is small, P_e can be approximated as $P_e \approx \binom{n}{t+1} p^{t+1}$. So, we have

$$P_e \approx \binom{n}{4} 0.01^4 \approx \left(\frac{n^4}{4!}\right) 0.01^4 = \left(\frac{(n/100)^4}{4!}\right).$$

Setting $P_e = 10^{-6}$, we get $n \approx \left(\frac{100^4 \cdot 24}{100^3}\right)^{\frac{1}{4}} \approx 7$. If we set $n = 7$, $p = 0.01$ in (13), we get $P_e \leq 3.417 \times 10^{-7}$ which is well below the target.

Remark: Note that the channel error probability of $p = 0.01$ is reduced to 10^{-6} by coding!

References

[1] T. K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms.* John Wiley & Sons, Inc., 2005.