Last lecture we began our study of error-correcting codes with the Hamming code as our primary example. Today we will discuss linear codes, of which the Hamming code is an example, and introduce both dual codes and the Singleton bound.

1 Hamming Code Recap

The Hamming code, parameterized by r, is a $[2^r-1, 2^r-1-r, 3]$ code, using the $[n, k, d]$ notation: n is the block length of the code, i.e. number of bits of the code word; k is the dimension of the code, i.e. the length of the message string; and d is the distance of the code, the minimum Hamming distance between any two codewords. Taking $r = 3$ gives us the usual 7-bit Hamming code, defined in terms of the parity check matrix H as

$$C_{\text{Ham},r=3} = \left\{ c \in \{0,1\}^7 : \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_7 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right\}. \tag{2.1}$$

Operations are done over the field $\mathbb{F}_2$, where we can do standard linear algebra over bits. The codewords are column linear dependencies of H. The parity check matrix H for generic r Hamming codes is given by stacking the columns of all non-zero bit strings of length r and is of size $r \times (2^r - 1)$. Consider a single bit flip error on bit i of codeword $c \in C_{\text{Ham},r}$. This error e_i is simply an $2^r - 1$ bit vectors of all zeros except a single 1 in position i. The corrupted bitstring is $y = c + e_i$, so $Hy = H(c + e_i) = Hc + He_i = He_i$. This selects the i^{th} column of H, which is a binary representation of i. Thus, as we saw last time, this code can correct for single bit-flip errors.

2 Code Distance

The distance of a code $C \subseteq \Sigma^n$ is $d(C) = \min_{c \neq c' \in C} \text{Ham Dist}(c, c')$, where the Hamming distance is the number of positions the two codewords differ. The code distance can geometrically understood in terms of a "packing radius" τ as shown in Fig. 2.1.

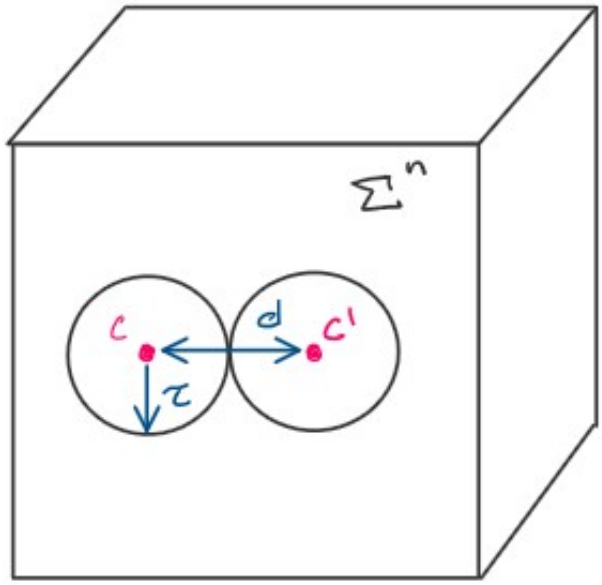

Figure 2.1: Geometric interpretation of the packing radius. The code C lives in the space Σ^n. Codewords c, c' is depicted by red dots. Around each codeword is a ball of radius τ; $B_n(c, \tau) = \{x \in \Sigma^n : \text{Ham Dist}(c, x) \leq \tau\}$. Two codewords are separated by at least the distance d.

Definition 2.1. The **packing radius** $\tau = \lfloor \frac{d-1}{2} \rfloor$ is the largest radius such that balls of radius τ around codewords are disjoint.

The error detection and correction properties of a code can be understood in terms of its distance. The following are equivalent for a code $C \subset \Sigma^n$.

1. C has distance $d \geq 2 * e + 1$.
2. C can correct up to e errors.
3. C can detect up to $2e$ errors.
4. C can correct up to $2e$ erasures.

The detection of $2e$ errors can be understood from our geometric intuition, as valid keywords are at least d changes apart; $2e < d$ errors is not sufficient to start at a valid codeword and end at another one. Correction of $2e$ erasures can be viewed as an errors with the location of the error specified. Thus we simply need to find the valid codewords from filling these $2e$ vacancies. There can only exist one valid codeword; otherwise multiple codewords would be related by $2e < d$ changes. Finally, consider the correction of e errors. Intuitively, length n bitstrings within a ball of radius τ can be indentified unambiguously with the codeword at ball center. We justify this with the triangle inequality. Given corrupted bit string y and codewords c, c'. Suppose that both c, c' can be reached by e changes to y, which is the situation when we cannot correct e errors. Then the Hamming weight $\Delta(c, c') \leq \Delta(c, y) + \Delta(c', y) \leq 2e$, which violates the distance of the code. Thus a distance $d = 2e + 1$ code can correct up to e errors. In the above analysis, we considered odd distance codes. Even distance codes are similar, given the definition of the packing radius.

Exercise 2.2. Prove that $C \subset \Sigma^n$ can correct e errors and s erasures if $2e + s < d(C)$.

Intuitively, errors are twice as expensive as erasures. As a hint, consider dropping the positions of the erasures. How does this affect the distance of the code?

A few comments before moving on:

1. We assumed the worst-case situation on the relation between errors, erasures, and distance. In Fig. 2.1, going half the distance directly towards another codeword is the hardest scenario for correction. However, in high dimension, i.e. large block length n, this is very unlikely. The "Shannon (probabilistic) perspective" considers how many errors can be corrected with high probability, rather than with certainty as we have been doing; on average more errors than half the distance can be corrected, typically on the order of the distance.

2. Even for worst case errors, the "Hamming perspective", we can *list decode* many more errors than those we can exactly correct, typically $\mathcal{O}(d)$. List decoding returns a list (of parameterized length) of possible codewords given a corrupted error string.

3 Linear Codes

Definition 2.3. A linear code $C \subset \mathbb{F}_q^n$ is a subspace of $\mathbb{F}_q^n$.

General codes have no prevailing structure or relation between codewords, so all codewords and the mapping to messages must be stored. The dimension of a code C can be written in terms of its rate R, $|C| = q^{Rn}$, so as n grows, assuming a finite rate code, this becomes prohibitively expensive. Linear codes avoid this and can be represented much more efficiently by a basis. Such codes are specified by a generator matrix G.

$$G = \begin{bmatrix} | & | & \dots & | \\ g_1 & g_2 & \dots & g_k \\ | & | & \dots & | \end{bmatrix} \in \mathbb{F}_q^{n \times k} \tag{2.2}$$

The vectors $g_i \in C$ form a basis of the code. The set of codewords is the space of the columns, $c \in \{Gx : x \in \mathbb{F}_q^k\}$. A code can also be specified by a parity check matrix $H \in \mathbb{F}_q^{n-k \times n}$, as in done in Eqn. 2.1.

$$C = \{c \in \mathbb{F}_q^n : Hc = 0\} = \mathrm{Ker}(H) \tag{2.3}$$

We see immediately that $HG = 0$. This is analogous to specifying a plane in Euclidean space either by a basis (G) or a normal vector (H). Every generator matrix has a "standard" form, where $G = [I_k|P]$ for $P \in \mathbb{F}_q^{n-k \times k}$, and a code with such a G is called a "systematic code". Given a generic G, a equivalent standard form G' can be found via standard row-reduction techniques. The parity check matrix H' of G' is $H' = [-P|I_{n-k}]$ such that $H'G' = 0^{n-k \times k}$ (Note that since we use a different convention for G as a column matrix than most texts, the expression for H' is different than that found in literature).

Example 2.4. Given a parity check matrix for $C_{\mathrm{Ham},r=3}$, find a generator matrix G.

The parity check matrix is a 3×7 dimensional matrix, with columns being the binary representation of $1-7$. We can choose $n-k$ linearly independent columns of H (highlighted in red below) to represent the checks and use the remaining k as message bits.

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{2.4}$$

The highlighted columns correspond to entries c_1, c_2, c_4 in codewords c, which we call "check bits" (even if they aren't in fact bits). From each row, we can read off a condition for these entries.

$$c_1 = c_3 + c_5 + c_7 \tag{2.5}$$
$$c_2 = c_3 + c_6 + c_7 \tag{2.6}$$
$$c_4 = c_5 + c_6 + c_7 \tag{2.7}$$

Thus the "message bits" c_3, c_5, c_g, c_7 are directly copies into the code word and three check bits are appended. This leads to the desired generator matrix.

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

One can check that $HG = 0$ as desired. ■

The distance of a code is defined as the minimum Hamming distance between any two codewords c, c'. Linear codes allow us to simplify this, as $c - c' \in C$.

Definition 2.5. The distance of a linear code C is $\min_{c \neq 0} \text{Ham Wt}(c) = \min_{c \neq 0} |\{i : c_i \neq 0\}|$.

4 Singleton Bound

There is an inherent tension between the distance $d(C)$ (relative distance $\delta(C) = d(C)/n$) and code dimension k (rate $R(C) = k/n$). This is quantified by the Singleton bound.

Lemma 2.6. *Singleton (Pigeonhole) Bound: Suppose $C \subset \mathbb{F}_q^n$. Then $|C| \leq q^{n-d(c)+1}$.*

Proof: Define the projection map $C \to \mathbb{F}_q^{n-d(C)+1}$ as $(c1, c2, \ldots, c_n) \mapsto (c1, c2, \ldots, c_{n-d(C)+1})$. Since we are erasing fewer than $d(C)$ bits, this map is one to one, given our earlier discussion about the correction of errors. But then the code dimension must be less than the number of possible strings of length $n - d(C) + 1$. Thus $|C| \leq q^{n-d(C)+1}$.

Reed-Solomon codes saturate these bounds but require $q = \mathcal{O}(n)$.

5 Dual of a code

Definition 2.7. The dual of a code C is $C^\perp = \{y \in \mathbb{F}_q^n : y \cdot c = 0 \, \forall c \in C\}$.

Note that $C^\perp$ is also a subspace of $\mathbb{F}_q^n$ and thus also a linear code. This dual code is the row-span(H) for any parity check matrix of C. Given a code C with generator matrix G and parity check matrix H, the dual code $C^\perp$ has generator matrix $H^\top$ and parity check matrix $G^\top$. Additionally $(C^\perp)^\perp = C$ and that $\dim(C^\perp) = n - k$. Thus $\dim(C) + \dim(C^\perp) = n$.

Over R, $C \cap C^\perp = \{0\}$, but over finite fields, the intersection can contain more than the zero string. In fact, the following are possibilities: (1) $C \subseteq C^\perp$; C is "self-orthogonal" and (2) $C = C^\perp$; C is "self-dual."

The dual of the Hamming code is the Simplex code. The generator matrix $G' = H^\top$ for H from C_{Ham}, so the rows are all non-zero length r vectors. The dimension of $C^\perp$ is r, so the rate $R(C^\perp) = r/(2^r - 1) \to 0$. The rate of C and its dual $C^\perp$ is related by $r(C) + r(C^\perp) = 1$. So $R(C) = 1 - R(C) \to 1$.

There is no generic relation between the distance of a code and it's dual, but for the Hamming and Simplex codes, $d(C_{\text{Ham}}) = 3$ and $d(C^\perp) = 2^{r-1} > n/2$. Thus the Hamming code has asymptotically optimal rate but relative distance of zero. The Simplex code has the opposite properties. Codes with both non-zero rate R and relative distance δ will be the subject of future lectures.