1 Hamming distance

- The Hamming distance between two strings of equal length, \( d_H(u, v) \), is defined as the number of positions at which the corresponding symbols are different.

- Triangular Inequality of Hamming distance:
  \[
  d_H(u, v) \leq d_H(u, w) + d_H(v, w).
  \]

- The Hamming weight of a string, \( wt_H(u) \), is the number of symbols that are different from the zero-symbol of the alphabet used, i.e.
  \[
  wt_H(u) = \sum_{i} |u_i \neq 0|.
  \]

  It is thus equivalent to the Hamming distance from the all-zero string of the same length.

- When the alphabet is binary, given two binary string \( u, v \) of equal length, we have
  \[
  d_H(u, v) = wt(u) + wt(v) - 2(u \cdot v),
  \]

  where ‘\( \cdot \)’ denotes inner product, by treating the components of \( u \) and \( v \) as integer 0 and 1.

Example 1: Given \( u = (11110000) \) and \( v = (00011110) \), we have \( wt(u) = wt(v) = 5 \), \( u \cdot v = 2 \) and \( d_H(u, v) = 5 + 5 - 4 = 6 \).

2 The Minimum Distance of a Code

- The minimum distance of a code \( C \) is defined as the smallest Hamming distance between two distinct codewords in \( C \). Specifically,
  \[
  d(C) = \min \{ d_H(c, c') : c, c' \in C, c \neq c' \}. \]

- The notation \( (n, M, d)_q \)-code \( C \) indicates that the code \( C \) is a \( q \)-ary code with length \( n \), size \( M \) and minimum distance \( d \). For binary code, i.e. \( q = 2 \), we simply write \( (n, M, d) \)-code.

Example 2: The rows of the following matrix are the codewords of a \((11,12,6)_2\)-code.
Matlab program: Calculate the Hamming distance of a code $C$ using Matlab.
Matlab codes can be found in https://piazza.com/class/isgy6spmvum3ba?cid=15.

**Theorem 1** (Error Correction). An $(n, M, d)_q$-code $C$ can correct $t$ errors if $d \geq 2t + 1$.

**Proof** Suppose codeword $c$ is transmitted and there are $t$ errors, for some integer $t$ satisfying $2t + 1 \leq d$.

Denote the received sequence by $y$. We have

$$d_H(c, y) = t.$$  \hfill (5)

The decoder we use is “nearest neighbor decoder”:

$$\text{Dec}(y) = \arg \min_{u \in C} d_H(u, y).$$  \hfill (6)

Suppose that there is decoding error, say codeword $c'$ is decoded erroneously, i.e., $c' \neq c$, and

$$d_H(c', y) = \min_{u \in C} d_H(u, y).$$  \hfill (7)

Since $d_H(c, y) = t$, we must have $d_H(c', y) \leq t$. Then, we get

$$d \leq d_H(c, c') \leq d_H(c, y) + d_H(c', y) \leq t + t = 2t,$$

where (8) follows since $c \neq c'$ and the minimum distance of code $C$ is $d$; (9) follows since Hamming distance satisfies Triangular Inequality.

Notice that (10) is a contradiction to our assumption that $d \geq 2t + 1$. Thus, given $d \geq 2t + 1$, any $t$ errors can be corrected. \hfill $\square$

**Theorem 2** (Error Detection). An $(n, M, d)_q$-code $C$ can detect any $s$ errors if $s \leq d - 1$.

**Proof** Suppose that a codeword $c \in C$ is transmitted, $s$ errors occur, and $y$ is received. The decoder we use is

$$\text{Dec}(y) = \begin{cases} c' & \exists c' \in C, \text{ s.t. } c' = y; \\ \text{error} & \text{otherwise.} \end{cases}$$  \hfill (11)

If there is an undetectable error, then we have $c' = y$ and $c' \neq c$. That is

$$s = d_H(c, y) = d_H(c, c') \geq d.$$  \hfill (12)

Thus, the decode could detect any $d - 1$ errors. \hfill $\square$

**Theorem 3** (Erasure Correction). An $(n, M, d)_q$-code $C$ can recover $r$ erasures if $r \leq d - 1$.

**Proof** Notation: For $J \subseteq \{1, 2, ..., n\}$, let $u_J \triangleq (u_j, j \in J)$

Example: If $y = (0, 1, 1, 1, 0, 0, 1)$ and $J = \{1, 2, 5, 6, 7\}$, then $u_J = (0, 1, 0, 0, 1)$.

Suppose codeword $c$ is transmitted, $r$ erasures occur and $y$ is received. Let $J$ be the set of indices of the unerased symbol. We note that $|J| = n - r$. 

2
The decoder we use is
\[
\text{Dec}(y) = \begin{cases} 
  c' & \exists \text{ a unique } c' \in C, \text{ s.t. } c'_J = y_J; \\
  \text{error} & \text{otherwise.}
\end{cases} 
\] (13)

If there is decoding error, then the decoder’s output \(\text{Dec}(y) = c'\) is a codeword different from the transmitted codeword \(c\), satisfying \(c'_J = c_J\). Since the components of codewords \(c\) and \(c'\) with indices in \(J\) are the same, we get
\[
d_H(c, c') \leq n - |J| = n - (n - r) = r. 
\] (14)

This contradicts the assumption that the Hamming distance between two distinct codewords is at least \(d\). Thus, the decoder can recover any \(d - 1\) erasures.

Example 3: Using the \((11, 12, 6)\)-code in Example 2, we can
\begin{itemize}
  \item correct 2 errors,
  \item detect 5 errors, or
  \item correct 5 erasures.
\end{itemize}

Exercises: Show that the converses of the above three theorems hold.

1. If we can design an error-correcting decoder for an \((n, M, d)_q\)-code that can correct any \(t\) errors, then \(d \geq 2t + 1\).
2. If we have an error-detecting decoder for an \((n, M, d)_q\)-code that can detect any \(s\) errors, then \(d \geq s + 1\).
3. If we have an erasure-correcting decoder for an \((n, M, d)_q\)-code that can correct any \(r\) erasures, then \(d \geq r + 1\).

Theorems 1 to 3, together with the above exercises, imply that an \((n, M, d)_q\) code can

1. correct up to \(\left\lfloor (d - 1)/2 \right\rfloor\) errors, but there exists an uncorrectable error pattern with \(\left\lfloor (d - 1)/2 \right\rfloor + 1\) errors.
2. detect up to \(d - 1\) errors, but there exists an undetectable error pattern with \(d\) errors.
3. recover up to \(d - 1\) erasures, but there exists an unrecoverable erasure pattern consisting of \(d\) erasures.