Syndrome decoding: example

An \((8, 4)\) binary linear block code \(C\) is defined by systematic matrices:

\[
H = \begin{bmatrix}
1 & 0 & 0 & 0 & | & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & | & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & | & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & | & 1 & 1 & 1 & 0 \\
\end{bmatrix} \quad \implies \quad G = \begin{bmatrix}
0 & 1 & 1 & 1 & | & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & | & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & | & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & | & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

Consider two possible messages:

\[
m_1 = [0 \ 1 \ 1 \ 0] \quad \quad m_2 = [1 \ 0 \ 1 \ 1] \\
\]

\[
c_1 = [0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0] \quad \quad c_2 = [0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1] \\
\]

Suppose error pattern \(e = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0]\) is added to both codewords.

\[
r_1 = [0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0] \quad \quad r_2 = [0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1] \\
\]

\[
s_1 = [1 \ 0 \ 1 \ 1] \quad \quad s_2 = [1 \ 0 \ 1 \ 1] \\
\]

Both syndromes equal column 6 of \(H\), so decoder corrects bit 6.

\(C\) is an expanded Hamming code with weight enumerator \(A(x) = 1 + 14x^4 + x^8\).
Standard array

Syndrome table decoding can also be described using the standard array.

The standard array of a group code $C$ is the coset decomposition of $F^n$ with respect to the subgroup $C$.

$$
\begin{array}{|c|cccc|}
\hline
0 & c_2 & c_3 & \cdots & c_M \\
\hline
 e_2 & c_2 + e_2 & c_3 + e_2 & \cdots & c_M + e_2 \\
 e_3 & c_2 + e_3 & c_3 + e_3 & \cdots & c_M + e_3 \\
 \vdots & \vdots & \vdots & \ddots & \vdots \\
 e_N & c_2 + e_N & c_3 + e_N & \cdots & c_M + e_N \\
\hline
\end{array}
$$

- The first row is the code $C$, with the zero vector in the first column.
- Every other row is a coset.
- The $n$-tuple in the first column of a row is called the coset leader. We usually choose the coset leader to be the most plausible error pattern, e.g., the error pattern of smallest weight.
The systematic generator and parity-check matrices for a \((6, 3)\) LBC are

\[
G = \begin{bmatrix}
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 0
\end{bmatrix} \quad \Rightarrow \quad H = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

The standard array has 6 coset leaders of weight 1 and one of weight 2.

<table>
<thead>
<tr>
<th>000000</th>
<th>001110</th>
<th>010101</th>
<th>011011</th>
<th>100011</th>
<th>101101</th>
<th>110110</th>
<th>111000</th>
</tr>
</thead>
<tbody>
<tr>
<td>000001</td>
<td>001111</td>
<td>010100</td>
<td>011010</td>
<td>100010</td>
<td>101100</td>
<td>110111</td>
<td>111001</td>
</tr>
<tr>
<td>000010</td>
<td>001100</td>
<td>010111</td>
<td>011001</td>
<td>100001</td>
<td>101111</td>
<td>110100</td>
<td>111010</td>
</tr>
<tr>
<td>000100</td>
<td>001010</td>
<td>010001</td>
<td>011111</td>
<td>100111</td>
<td>101001</td>
<td>110010</td>
<td>111100</td>
</tr>
<tr>
<td>001000</td>
<td>000110</td>
<td>011101</td>
<td>010011</td>
<td>101011</td>
<td>100101</td>
<td>111110</td>
<td>110000</td>
</tr>
<tr>
<td>010000</td>
<td>011110</td>
<td>000101</td>
<td>001011</td>
<td>110011</td>
<td>111101</td>
<td>100110</td>
<td>101000</td>
</tr>
<tr>
<td>100000</td>
<td>101110</td>
<td>110101</td>
<td>111011</td>
<td>000011</td>
<td>001101</td>
<td>010110</td>
<td>011000</td>
</tr>
<tr>
<td>001001</td>
<td>000111</td>
<td>011100</td>
<td>010010</td>
<td>101010</td>
<td>100100</td>
<td>111111</td>
<td>110001</td>
</tr>
</tbody>
</table>

See http://www.stanford.edu/class/ee387/src/stdarray.pl for the short Perl script that generates the above standard array. This code is a \emph{shortened} Hamming code.
Standard array: decoding

An \((n, k)\) LBC over GF\((Q)\) has \(M = Q^k\) codewords.

Every \(n\)-tuple appears exactly once in the standard array. Therefore the number of rows \(N\) satisfies

\[
MN = Q^n \implies N = Q^{n-k}.
\]

All vectors in a row of the standard array have the same syndrome. Thus there is a one-to-one correspondence between the rows of the standard array and the \(Q^{n-k}\) syndrome values.

Decoding using the standard array is simple: decode senseword \(r\) to the codeword at the top of the column that contains \(r\).

The decoder subtracts the coset leader from the received vector to obtain the estimated codeword.

The \textit{decoding region} for a codeword is the column headed by that codeword.
### Standard array and decoding regions

<table>
<thead>
<tr>
<th></th>
<th>codewords</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><strong>shells of radius 1</strong></td>
</tr>
<tr>
<td>wt 1</td>
<td><strong>shells of radius 2</strong></td>
</tr>
<tr>
<td>wt 2</td>
<td><strong>shells of radius t</strong></td>
</tr>
<tr>
<td>wt t</td>
<td><strong>vectors of weight &gt; t</strong></td>
</tr>
<tr>
<td>wt &gt;t</td>
<td><strong>vectors of weight &gt; t</strong></td>
</tr>
</tbody>
</table>
Bounds on minimum distance

The minimum distance of a block code is a conservative measure of the quality of an error control code.

- A large minimum distance guarantees reliability against random errors.
- However, a code with small minimum distance may be reliable, if the probability of sending codewords with nearby codewords is small.

We use minimum distance as the measure of a code’s reliability because:

- A single number is easier to understand than a weight/distance distribution.
- The guaranteed error detection and correction ability are
  - detection: \( e = d^* - 1 \)
  - correction: \( t = \left\lfloor \frac{1}{2} (d^* - 1) \right\rfloor \)
- Algebraic codes covered in the course are limited by minimum distance. Their decoders cannot correct more than \( t \) errors even if there is only one closest codeword.
The Hamming bound for a \((n, k)\) block code over \(Q\)-ary channel alphabet:

- A code corrects \(t\) errors iff spheres of radius \(t\) around codewords do not overlap. Therefore

\[
Q^k = \text{number of codewords} \leq \frac{\text{volume of space}}{\text{volume of sphere of radius } t} = \frac{Q^n}{V(Q, n, t)},
\]

where \(V(Q, n, t)\) is the "volume" (number of elements) of a sphere of radius \(t\) in Hamming space of \(n\)-tuples over a channel alphabet with \(Q\) symbols:

\[
V(Q, n, t) = 1 + \binom{n}{1}(Q-1) + \binom{n}{2}(Q-1)^2 + \cdots + \binom{n}{t}(Q-1)^t
\]

- Rearranging the inequality gives a lower bound on \(n - k\) and thus an upper bound on rate \(R\):

\[
Q^{n-k} \geq V(Q, n, t) \implies n - k \geq \log_Q V(Q, n, t)
\]

\[
R \leq 1 - \frac{1}{n} \log_Q \left(1 + \binom{n}{1}(Q-1) + \binom{n}{2}(Q-1)^2 + \cdots + \binom{n}{t}(Q-1)^t\right)
\]
Hamming bound: example

A wireless data packet contains 192 audio samples, 16 bits for each of two channels. The number of information bits is \(192 \cdot 2 \cdot 16 = 6144\).

The communications link is a binary symmetric channel with raw error rate \(10^{-3}\). How many check bits are needed for reliable communication?

<table>
<thead>
<tr>
<th>(t)</th>
<th>(n - k)</th>
<th>(n)</th>
<th>Rate</th>
<th>(\text{Pr}{&gt; t \text{ errors}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>105</td>
<td>6249</td>
<td>0.983</td>
<td>(5.4 \times 10^{-02})</td>
</tr>
<tr>
<td>12</td>
<td>123</td>
<td>6267</td>
<td>0.980</td>
<td>(1.2 \times 10^{-02})</td>
</tr>
<tr>
<td>14</td>
<td>141</td>
<td>6285</td>
<td>0.978</td>
<td>(2.2 \times 10^{-03})</td>
</tr>
<tr>
<td>16</td>
<td>158</td>
<td>6302</td>
<td>0.975</td>
<td>(3.0 \times 10^{-04})</td>
</tr>
<tr>
<td>18</td>
<td>175</td>
<td>6319</td>
<td>0.972</td>
<td>(3.5 \times 10^{-05})</td>
</tr>
<tr>
<td>20</td>
<td>192</td>
<td>6336</td>
<td>0.970</td>
<td>(3.3 \times 10^{-06})</td>
</tr>
<tr>
<td>24</td>
<td>225</td>
<td>6369</td>
<td>0.965</td>
<td>(1.8 \times 10^{-08})</td>
</tr>
<tr>
<td>28</td>
<td>257</td>
<td>6401</td>
<td>0.960</td>
<td>(5.5 \times 10^{-11})</td>
</tr>
<tr>
<td>32</td>
<td>288</td>
<td>6432</td>
<td>0.955</td>
<td>(1.0 \times 10^{-13})</td>
</tr>
</tbody>
</table>

The Hamming bound shows that more than 4\% redundancy is needed to achieve a reasonable bit error rate.
Other bounds on minimum distance

- **Plotkin upper bound for binary linear block codes** (homework exercise):

\[ d^* \leq \frac{n \cdot 2^{k-1}}{2^k - 1} \implies \delta = \frac{d^*}{n} \leq \frac{1}{2} \text{ for large } k. \]

\[ \delta = d^*/n \text{ is normalized minimum distance.} \]

- **McEliece-Rodemich-Rumsey-Welch (MRRW) upper bound.**

\[ R \leq H\left(\frac{1}{2} - \sqrt{\delta(1-\delta)}\right). \]

\( H \) is binary entropy function. MRRW bound is better than Hamming or Plotkin for some ranges of \( \delta \).

- **Varshamov-Gilbert lower bound** for binary block codes. If \( d^* < n/2 \) there then exists a code with minimum distance \( d^* \) and rate \( R \) satisfying

\[ R \geq 1 - \log_2 \left( \sum_{i=0}^{d-1} \binom{n}{i} \right) \approx 1 - H(d^*/n) = 1 - H(\delta). \]

For comparison, the Hamming bound is \( R \leq 1 - H(\delta/2) \).
The MRRW bound is stronger than Hamming bound except for high rates.

The Hamming bound is fairly tight for high rates. E.g., to correct 10 errors in 1000 bits, Hamming bound requires 78 check bits, but there exists a BCH code with 100 check bits.
Perfect codes

Definition: A block code is called perfect if every senseword is within distance $t$ of exactly one codeword.

Other definitions of perfect codes:

- Decoding spheres pack perfectly
- Have complete bounded-distance decoders
- Satisfy the Hamming bound with equality

There are only finitely many classes of perfect codes:

- Codes with no redundancy ($k = n$)
- Repetition codes with odd blocklength: $n = 2m + 1, k = 2m, t = m$
- Binary Hamming codes: $n = 2^m - 1, n - k = m$
- Nonbinary Hamming codes: $n = (q^m - 1)/(q - 1), n - k = m, q > 2$
- Binary Golay code: $q = 2, n = 23, k = 12, t = 3$
- Ternary Golay code: $q = 3, n = 11, k = 6, t = 2$

Golay discovered both perfect Golay codes in 1949—a very good year for Golay.
Quasi-perfect codes

Definition: A code is quasi-perfect if every $n$-tuple

- is within distance $t$ of at most one codeword, and
- is within distance $t+1$ of at least one codeword.

In other words, a code is quasi-perfect if

- spheres of radius $t$ surrounding codewords do not overlap, while
- spheres of radius $t+1$ cover the space of $n$-tuples.

Examples of quasi-perfect codes:

- Repetition codes with even blocklength
- Expanded Hamming and Golay codes with overall parity-check bit

Exercise: Show that expurgated Hamming codes (obtained by adding an overall parity-check equation) are not quasi-perfect.