Lectures 22 and 23.

BCH codes.

Recall that these are $q$-ary codes defined by $d \times n$ parity check matrices of the form

$$H = \begin{pmatrix}
    1 & 1 & \ldots & 1 \\
    a_1 & a_2 & \ldots & a_n \\
    a_1^2 & a_2^2 & \ldots & a_n^2 \\
    \vdots & \vdots & \ddots & \vdots \\
    a_1^{d-1} & a_2^{d-1} & \ldots & a_n^{d-1}
\end{pmatrix},$$

where $1 \leq d + 1 \leq n \leq q$ and $a_1, \ldots, a_n$ are distinct elements of $F_q$. We showed that the resulting $q$-ary code has length $n$, dimension $n - d$ and minimal distance $d + 1$.

Note that the case where $d = 0$ corresponds to the “empty” parity check matrix. This means that our code contains every $q$-ary word on length $n$ (no parity check conditions). The dimension of this code is $n - d = n$ and the minimum distance is $d + 1 = 1$.

Last time we focused on a particular example with $q = 11$, $n = 10$, and $a_i = i$ for $i = 1, 2, \ldots, 10$. The parity check matrix for our code $C$ is thus

$$H = \begin{pmatrix}
    1 & 1 & \ldots & 1 \\
    1 & 2 & \ldots & 10 \\
    1^2 & 2^2 & \ldots & 10^2 \\
    1^3 & 2^3 & \ldots & 10^3
\end{pmatrix},$$

The minimal distance of this code is 5. A decoding algorithm correcting up to two errors is described in Example 11.3 in the book.
Examples

Last time I illustrated this algorithm by decoding the received vector as \( \mathbf{y} = (2, 1, 2, 0, 0, 0, 0, 0, 0, 0) \). Here the error-locator polynomial had two distinct non-zero roots in \( F_{11} \), so we were in case (iii) of the algorithm on p. 130. Now I would like to present two further examples, illustrating other cases in the decoding algorithm.

**Example:** Suppose the received vector is \( \mathbf{y} = (1, 1, 1, 0, 0, 0, 0, 9) \). Compute the syndrome \( (s_1, s_2, s_3, s_4) = S(\mathbf{y}) = \mathbf{y} \cdot H^T \), where \( s_1 = 2, s_2 = 1, s_3 = 6, \) and \( s_4 = 3 \).

The error-locator polynomial is \( P t^2 + Q t + R \), where

\[
P = s_2^2 - s_1 s_3 = 1 - 12 = 0, \\
Q = s_1 s_4 - s_2 s_3 = 6 - 6 = 0, \text{ and} \\
R = s_3^2 - s_2 s_4 = 36 - 3 = 0.
\]

Here all computations are performed in \( F_{11} \). Since \( P = Q = 0 \), we are in Case (ii) of the decoding algorithm on p. 130. We assume one error of magnitude \( a \) in position \( i \), i.e., an error vector of the form

\[
\mathbf{e} = (0, \ldots, 0, a, 0, \ldots, 0).
\]

Now \( S(\mathbf{e}) = S(\mathbf{y}) \), so

\[
(a, ai, ai^2, ai^3) = (s_1, s_2, s_3, s_4) = (2, 1, 6, 3).
\]

Equating the first components, we see that \( a = 2 \). Equating the second components, we obtain \( 2i = 1 \), so \( i = 2^{-1} = 6 \) in \( F_{11} \). Use the last two components to check our answer: \( ai^2 = 2 \cdot 6^2 = 6 \) and \( ai^3 = 2 \cdot 6^3 = 2 \cdot 6 \cdot 3 = 2 \cdot 7 = 3 \).

The error vector is \( \mathbf{e} = (0, 0, 0, 0, 0, 2, 0, 0, 0, 0) \). We decode \( \mathbf{y} \) as \( \mathbf{x} = \mathbf{y} - \mathbf{e} = (1, 1, 1, 0, 9, 0, 0, 0, 9) \).

**Another example:** Suppose the received vector is

\[
\mathbf{y} = (0, 0, 0, 0, 0, 0, 1, 1, 1).
\]

2
Computing the syndrome \((s_1, s_2, s_3, s_4) = S(y) = y \cdot H^T\), we find that \(s_1 = 3\), \(s_2 = 5\), \(s_3 = 3\), and \(s_4 = 8\).

The error-locator polynomial is \(Pt^2 +Qt + R\), where
\[
P = s_2^2 - s_1s_3 = 25 - 9 = 5,
\]
\[
Q = s_1s_4 - s_2s_3 = 24 - 15 = 9 = -2, \text{ and}
\]
\[
R = s_3^2 - s_2s_4 = 9 - 40 = 2.
\]
The discriminant of the error-locator polynomial, \(D = Q^2 - 4PR = 4 - 40 = -3 = 8\) is a complete square in \(F_{11}\); see the table of squares on p. 130. Thus we in case (iv) of the decoding algorithm: at least 3 errors occurred in transmission.

**A more general BCH code**

I will now discuss a decoding algorithm for a more general BCH code corresponding to the \(d \times n\)-parity check matrix
\[
A = \begin{pmatrix}
1 & 1 & \ldots & 1 \\
1 & 2 & \ldots & n \\
1^2 & 2^2 & \ldots & n^2 \\
\cdots & \cdots & \cdots & \cdots \\
1^{d-1} & 2^{d-1} & \cdots & n^{d-1}
\end{pmatrix},
\]
In our previous example \(q = 11\), \(n = 10\) and \(d = 4\); here I am assuming that \(q\) is an arbitrary prime number, and \(d = 2t\) is even, and \(n\) and \(q\) are arbitrary, subject to \(d < n < q\). The minimal distance of this code is \(d + 1 = 2t + 1\), so it will be capable of correcting up to \(t\) errors. I will now outline a decoding algorithm for this code. This decoding procedure will not be covered on the final exam. I am presenting it here, because it is very elegant, and because it sheds new light on the decoding procedure of Example 11.3 (which will be covered). At the end, I will return to the three
specific received vectors using the code of Example 11.3 and redo them from this new point of view.

Suppose the received word is \( \mathbf{y} = (y_1, \ldots, y_n) \). The syndrome of \( \mathbf{y} \) is \((s_1, \ldots, s_d)\), where
\[
s_i = y_1 1^{i-1} + y_2 2^{i-1} + \cdots + y_n n^{i-1}.
\] (1)

We want to match \( S(\mathbf{y}) \) to the syndrome of a hypothetical error vector \( \mathbf{e} \) with errors of magnitudes \( m_1, \ldots, m_t \) in positions \( p_1, \ldots, p_t \). Here we allow some of the magnitudes \( m_1, \ldots, m_t \) to be 0; this will simply mean that fewer than \( t \) errors occurred. Matching the syndrome for \( \mathbf{e} \) (on the left) with the syndrome for \( \mathbf{y} \) (on the right), we obtain the following system of equations:
\[
\begin{align*}
    m_1 + \cdots + m_t &= s_1 \\
    m_1 p_1 + \cdots + m_t p_t &= s_2 \\
    m_1 p_1^2 + \cdots + m_t p_t^2 &= s_3 \\
    & \vdots \\
    m_1 p_1^{2t-1} + \cdots + m_t p_t^{2t-1} &= s_{2t}
\end{align*}
\] (2)

Note that by our convention \( d = 2t \). Here \( s_1, \ldots, s_d = s_{2t} \) are known to us. We want to solve this system for \( m_1, \ldots, m_t \) and \( p_1, \ldots, p_t \). There are \( 2t \) equations in \( 2t \) variables, so it is reasonable to expect that this system is likely to have a solutions (at least for “many” or “most” choices of \( s_1, \ldots, s_{2t} \). However, finding a solution is likely to be difficult, because the equations in our system are non-linear.

Note also that this system can be continued indefinitely:
\[
\begin{align*}
    m_1 p_1^{2t} + \cdots + m_t p_t^{2t} &= s_{2t}, \\
    m_1 p_1^{2t+1} + \cdots + m_t p_t^{2t+1} &= s_{2t+1},
\end{align*}
\]

etc. The quantities on the right hand side \( s_{2t}, s_{2t+1}, \) etc. are no longer the components of the syndrome \( S(\mathbf{y}) \); nevertheless, they can be computed from the received word \( \mathbf{y} \) using formula (1).
To solve our system, we introduce the function
\[ \phi(x) = \frac{m_1}{1 - p_1 x} + \frac{m_2}{1 - p_2 x} + \cdots + \frac{m_t}{1 - p_t x}. \]
The connection to our system comes from the fact that
\[ \frac{m}{1 - px} = m(1 + px + p^2x^2 + p^3x^3 + \ldots) \]
Note that the series on the right hand side is just a formal power series. We are working in a finite field, so we cannot substitute particular values for \( x \) (other than 0) and expect any kind of convergence, the way one would we do in real or complex analysis. For us this power series will simply be a bookkeeping device, allowing us to neatly separate the powers of \( p \). Formal power series can be added together and, most importantly, multiplied, and we shall do this below. In particular, adding together the power series
\[ \frac{m_1}{1 - p_1 x} = m_1(1 + p_1 x + p_1^2 x^2 + p_1^3 x^3 + \ldots) \]
\[ \frac{m_2}{1 - p_2 x} = m_2(1 + p_2 x + p_2^2 x^2 + p_2^3 x^3 + \ldots) \]
\[ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \]
\[ \frac{m_t}{1 - p_t x} = m_t(1 + p_t x + p_t^2 x^2 + p_t^3 x^3 + \ldots) \]
we obtain
\[ \phi(x) = s_1 + s_2 x + s_3 x^2 + \cdots + s_{2t} x^{2t-1} + \ldots \quad (3) \]
Note that if we know \( m_1, \ldots, m_t, p_1, \ldots, p_t \), then we can write the terms in \( \phi(x) \) under a common denominator, as
\[ \phi(x) = \frac{a_1 + a_2 x + \cdots + a_t x^{t-1}}{1 + b_1 x + \cdots + b_t x^t}. \quad (4) \]
If we know \( A(x) = a_1 + a_2 x + \cdots + a_t x^{t-1} \) and \( B(x) = 1 + b_1 x + \cdots + b_t x^t \), then we know \( \phi(x) = A(x)/B(x) \), and we can recover \( m_1, \ldots, m_t, p_1, \ldots, p_t \) by decomposing \( \phi(x) = A(x)/B(x) \) into partial fractions. (More on this below.) So we may view passing from \( m_1, \ldots, m_t, p_1, \ldots, p_t \) to \( a_1, \ldots, a_t, b_1, \ldots, b_t \) as a change of variables (of sorts). It turns out that our original non-linear system in terms of \( m_1, \ldots, m_t, p_1, \ldots, p_t \) can be rewritten as a simpler system in \( a_1, \ldots, a_t, b_1, \ldots, b_t \), which can be solved in terms of linear algebra.

Here is what I mean. Multiplying both sides of (3) by \( B(x) = 1 + b_1 x + \cdots + b_t x^t \), we obtain
\[
a_1 + a_2 x + \cdots + a_t x^{t-1} = (1 + b_1 x + \cdots + b_t x^t) (s_1 + s_2 x + \cdots + s_{2t} x^{2t-1} + \ldots).
\]
Equating the coefficients of \( 1, x, \ldots, x^{2t-1} \) on both sides, we obtain
\[
\begin{align*}
a_1 &= s_1 \\
a_2 &= s_2 + s_1 b_1 \\
a_3 &= s_3 + s_2 b_1 + s_1 b_2 \\
\vdots \\
a_t &= s_t + s_{t-1} b_1 + s_{t-2} b_2 + \cdots + s_1 b_{t-1} \\
0 &= s_{t+1} + s_t b_1 + \cdots + s_1 b_t \\
0 &= s_{t+2} + s_{t+1} b_1 + \cdots + s_2 b_t \\
\vdots \\
0 &= s_{2t} + s_{2t-1} b_1 + \cdots + s_t b_t
\end{align*}
\] (5)
The last \( t \) equations (with 0 on the left hand side) form a system of linear equations with variables \( b_1, \ldots, b_t \) (we know all the coefficients). There are \( t \) equations and \( t \) unknowns, so in most cases we expect a unique solution. Once we know \( b_1, \ldots, b_t \), we can use the remaining \( t \) equations to solve for \( a_1, \ldots, a_t \).

Now we know \( \phi(x) \) (see (4)), and we can use a partial fractions expansion to find \( m_1, \ldots, m_t, p_1, \ldots, p_t \). A practical way to do this
is as follows. By our construction,

\[ B(x) = (1 - p_1 x) \ldots (1 - p_t x). \]

Its roots are \(1/p_1, 1/p_2, \ldots, 1/p_t\) are the inverses of the error positions. So, if \(B(x)\) has \(t\) distinct non-zero roots, \(r_1, \ldots, r_t\) in \(F_q\), then the error positions \(p_1, \ldots, p_t\) are simply the inverses of these roots \(r - 1^{-1}, \ldots, r_t^{-1}\). For this reason \(B(x)\) is called the error-locator polynomial. (Note a slight clash with the terminology we used in Example 11.3. The error-locator polynomial in that example actually had error locations as roots. On the other hand, the the roots of \(B(x)\), are inverses of the error locations in \(F_q\).)

Note also that finding roots of \(B(x)\) in \(F_q\) is not as difficult as finding real roots of polynomials with real coefficients, at least if \(q\) is reasonably small. The reason is that we can simply plug in all possible elements of \(F_q\) into \(B(x)\) and see which ones are roots.

Once we know the error positions \(p_1, \ldots, p_t\), our original system (2) becomes linear, and we can solve it for the magnitudes \(m_1, \ldots, m_t\). This way we we avoid computing \(A(x)\) or decomposing \(A(x)/B(x)\) into a sum of partial fractions. (Well, if we did look for such a partial fraction decomposition, we would have ended up with the same linear system. But at least conceptually we don’t need to worry about it here.)

What happens if the number of errors \(e\) is actually \(< t\)? This can manifest itself in one of two ways. One possibility is that the degree of \(B(x)\) may be lower than \(t\), i.e., \(b_t = 0\). Another is that \(B(x)\) has \(t\) distinct roots, whose inverses should be error positions, but some of the error magnitudes associated to these positions are 0. We will see an example of this phenomenon in Example 2 below.

What happens if \(B(x)\) is of degree \(e \leq t\) but has fewer than \(e\)
distinct roots in $F_q$? In this case we conclude that more that $t$ errors occurred.

**Back to our old examples**

Let us return to our examples, where $q = 11$, $n = 10$, $d = 4$ (and thus $t = 2$).

**Example 1.** Suppose the received vector is $y = (2, 1, 2, 0, 0, 0, 0, 0, 0, 0)$. We’ve computed the syndrome $(s_1, s_2, s_3, s_4) = S(y) = (5, -1, 2, 9)$. The last two equations in (5) assume the form

\[
0 = 2 - b_1 + 5b_2 \\
0 = 9 + 2b_1 - b_2
\]

Solving for $b_1$ and $b_2$, we obtain $b_1 = 7$ and $b_2 = 1$. Thus $B(x) = 1 + b_1x + b_2x^2 = 1 + 7x + x^2$. The discriminant of this polynomial is $D = 7^2 - 4 \cdot 1 \cdot 1 = 45 = 1$ is a complete square in $F_{11}$. Using the quadratic formula, we find that $B(x)$ has two distinct roots in $F_{11}$, $r_1 = 7$ and $r_2 = 8$. Thus the error positions are $p_1 = r_1^{-1} = 8$ and $p_2 = r_2^{-1} = 7$. This is consistent with our previous findings (recall that back then we denoted the error positions by $i$ and $j$.)

The rest is the same as before: we substitute $p_1 = 8 = -3$ and $p_2 = 7 = -4$ back into the system (2) and solve for the error magnitudes $m_1$ and $m_2$. We find that $m_1 = m_2 = 8$. Thus the error vector is

\[
e = (0, 0, 0, 0, 0, 8, 8, 0, 0, 0)
\]

and we decode $y$ and

\[
x = y - e = (2, 1, 2, 0, 0, 3, 3, 0, 0).
\]

**Example 2.** Here the received word is $y = (1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 9)$. 

We’ve computed the syndrome to be \((s_1, s_2, s_3, s_4) = S(y) = (2, 1, 6, 3)\). The last two equations in (5) assume the form
\[
0 = 6 + b_1 + 2b_2 \\
0 = 3 + 6b_1 + 1b_2
\]
Solve for \(b_1\) and \(b_2\): \(b_1 = 0\) and \(b_2 = -3\). Thus \(B(x) = 1 + b_1 x + b_2 x^2 = 1 - 3x^2\) has two roots, \(r_1 = 2\) and \(r_2 = -2\). The error positions are thus \(p_1 = 2^{-1} = 6\) and \(r_2 = (-2)^{-1} = -6 = 5\).

When we previously looked at this example, we found that there was only one error, in position 6. Why are we getting a second error position this time? It turns out that when we we substitute \(p_1 = 8 = -3\) and \(p_2 = 7 = -4\) back into the the system (2) and solve for the error magnitudes \(m_1\) and \(m_2\), we find that \(m_1 = 2\) and \(m_2 = 0\). So, there was only one error after all. The error vector is
\[
e = (0, 0, 0, 0, 2, 0, 0, 0, 0)
\]
and we decode \(y\) as
\[
x = y - e = (1, 1, 1, 0, 9, 0, 0, 0, 9).
\]
as before.

**Example 3.** Here the received word is
\[
y = (0, 0, 0, 0, 0, 0, 0, 1, 1, 1).
\]

We have already computed the syndrome
\[
(s_1, s_2, s_3, s_4) = S(y) = (3, 5, 3, 8).
\]
Our system (5) assumes the form
\[
a_1 = 3 \\
a_2 = 5 + 3b_1 \\
0 = 3 + 5b_1 + 3b_2 \\
0 = 8 + 3b_1 + 5b_2
\]
Solving the last two equations for $b_1$ and $b_2$, we obtain $b_1 = 4$ and $b_2 = -4$. Thus $B(x) = 1 + b_1 x + b_2 x^2 = 1 + 4x - 4x^2$. The discriminant of this polynomial is $D = 4^2 - 4 \cdot (-4) \cdot 1 = 10$ is not a complete square in $\mathbb{F}_{11}$ (see the table of squares on p. 130). We conclude that at least three occurred in transmission.