1 Error Correcting Codes

We will work with degree $d$ polynomials over the field $GF(p)$ of numbers mod $p$. This means that we consider polynomials of the form $f(x) = c_0 + c_1 x + \cdots + c_d x^d$, where the coefficients $c_0, c_1, \ldots, c_d$ are chosen from $GF(p)$.

The key property of degree $d$ polynomials that we will use is that given $d + 1$ pairs of values $(a_0, b_0), (a_1, b_1), \ldots, (a_d, b_d)$ (all from $GF(p)$), there is a unique degree $d$ polynomial $f(x)$ that takes on these values in the sense that $f(a_j) = b_j$ for $i = 0$ to $d$.

To construct this polynomial $f(x)$ we use Legendre interpolation: $f(x) = \sum_{j=0}^{d} b_j \prod_{k \neq j} (x - a_k) / (a_j - a_k)$

Here we may think of $\Delta_j(x) = \prod_{k \neq j} (x - a_k)$ as a “delta-function”, since it takes on value 1 at $a_j$ and 0 at $a_k$ for $k \neq j$.

To prove the uniqueness of this polynomial $f(x)$ we used the fact that any polynomial of degree $d$ has at most $d$ roots.

The coding Problem:

Suppose we wish to transmit a sequence of numbers $b_0, b_1, \ldots, b_d$ over a noisy communication channel. The numbers are over $GF(p)$, i.e. mod $p$. Each number that we choose to transmit over the communication channel has some chance of getting corrupted. What we would like to do is to encode the given sequence $b_0, b_1, \ldots, b_d$ into a longer sequence of numbers $c_0, c_1, \ldots, c_{n-1}$, and transmit this longer sequence over the noisy communication channel. The property of this encoding that we would like to ensure is that even if some constant fraction (say $1/4$) of the $c_j$’s are corrupted, the recipient can still reconstruct the original sequence $b_0, b_1, \ldots, b_d$.

The procedure for carrying out this encoding is very simple: Consider the unique polynomial $f(x)$ which takes on values $b_0, b_1, \ldots, b_d$ at the points $0, 1, \ldots, d$. i.e. $f(j) = b_j$ for $j = 0$ to $d$.

Now let $c_j = f(j)$ for $j = 0$ to $n - 1$.

Recovering from errors:

Suppose that there are $k$ errors in the transmitted numbers $c_0, c_1, \ldots, c_{n-1}$, i.e. the recipient got the sequence of numbers $f_0, f_1, \ldots, f_{n-1}$ instead, where $f_j \neq c_j$ for at most $k$ different $j$’s. Can the recipient recover the original sequence $b_0, b_1, \ldots, b_d$ despite these errors? This looks hard because the recipient does not even know which values are correct and which are erroneous. Nevertheless, if $k \leq \frac{n-d-1}{2}$ then it is possible to recover the original sequence $b_0, b_1, \ldots, b_d$. Notice that since there is no a priori constraint on how much larger $n$ can be
compared to $d$, this bound on $k$ can be made arbitrarily close to $n/2$. i.e. we can recover from nearly 50% of the data being erroneous.

Notice that since at least $\frac{n+d+1}{2}$ of the transmitted numbers are correct, the original polynomial $f(x)$ agrees with at least $\frac{n+d+1}{2}$ of the values that the recipient gets. Now, we claim that any degree $d$ polynomial $g(x)$ that agrees with any $\frac{n+d+1}{2}$ of the received values (not necessarily the uncorrupted ones), must actually be the correct polynomial $f(x)$. To see this notice that only $\frac{n-d-1}{2}$ of the received values are corrupted. So among any choice of $\frac{n+d+1}{2}$ of the received numbers, at least $\frac{n+d+1}{2} - \frac{n+d+1}{2} = d + 1$ must be uncorrupted. Now, since a degree $d$ polynomial is uniquely determined by its values at $d+1$ points, it follows that any degree $d$ polynomial consistent with these values must in fact be the polynomial $f(x)$.

**The Berlekamp-Welsch decoder:**

How do we efficiently reconstruct a polynomial that is consistent with at least $\frac{n+d+1}{2}$ of the received numbers? There is an efficient (and magical) algorithm called the Berlekamp-Welsch decoding algorithm. Details to follow.