The Berlekamp-Welch Decoder

This section presents the solution to the following problem first introduced by Berlekamp and Welch as part of a novel method for decoding Reed-Solomon codes.

Problem 4

`Given :` *m pairs of points $(x_i, s_i) \in F \times F$ such that there exists a polynomial K of degree at most d such that for all but k values of i, $s_i = K(x_i)$, where $2k + d < m$.*

`Question :` *Find K*

NOTA: unfortunately, k and d are the opposite of what we have used so far !!!! Also, m stands for n...

The Berlekamp-Welch Decoder

Consider the following set of equations:

$$\exists W, K \quad \deg(W) \leq k, \deg(K) \leq d, W \neq 0, \text{ and } \forall i \quad W(x_i) * s_i = W(x_i) * K(x_i) \tag{1}$$

Any solution W, K to the above system gives a solution to Problem 4. (Notice that we can cancel W from both sides of the equation to get $s_i = f(x_i)$, except when $W(x_i) = 0$, but this can happen at most k times.)

The Berlekamp-Welch Decoder

Conversely, any solution K to Problem 4 also gives a solution to the system of equations .

(Let $B = \{x_i | s_i \neq f(x_i)\}$. Let $W(z)$ be the polynomial $\prod_{x \in B}(z - x)$. W, K form a solution to the system 1.)

Thus the problem can be reduced to the problem of finding polynomials K and W that satisfy (1).

The Berlekamp-Welch Decoder

Now consider the following related set of constraints

$$\exists W, N \quad \deg(W) \leq k, \deg(N) \leq k + d, W \neq 0, \text{ and } \forall i \quad W(x_i) * s_i = N(x_i) \tag{2}$$

If a solution pair N, W to (2) can be found that has the additional property that W divides N, then this would yield K and W that satisfy (1). Berlekamp and Welch show that all solutions to the system (2) have the same N/W ratio (as rational functions) and hence if equation (2) has a solution where W divides N, then any solution to the system (2) would yield a solution to the system (1). The following lemma establishes this invariant.

The Berlekamp-Welch Decoder

Lemma 6 *Let N, W and L, U be two sets of solutions to (2). Then $N/W = L/U$.*

Proof: For i, $1 \leq i \leq m$, we have

$$L(x_i) = s_i * U(x_i) \quad \text{and} \quad N(x_i) = s_i * W(x_i)$$

$$\Rightarrow L(x_i) * W(x_i) * s_i = N(x_i) * U(x_i) * s_i$$

$$\Rightarrow L(x_i) * W(x_i) = N(x_i) * U(x_i) \qquad \text{(by cancellation)}$$

(Cancellation applies even when $s_i = 0$ since that implies $N(x_i) = L(x_i) = 0$.) But both $L * W$ and $N * U$ are polynomials of degree at most $2k + d$ and hence if they agree on $m > 2k + d$ points they must be identical. Thus $L * W = N * U \ \Rightarrow\ L/U = N/W$ $\square$

The Berlekamp-Welch Decoder

All that remains to be shown is how one obtains a pair of polynomials W and N that satisfy (2). To obtain this, we substitute unknowns for the coefficients of the polynomials i.e., let $W(z) = \sum_{j=0}^{k} W_j z^j$ and let $N(z) = \sum_{j=0}^{k+d} N_j z^j$. To incorporate the constraint $W \neq 0$ we set $W_k = 1$. Each constraint of the form $N(x_i) = s_i * W(x_i)$, $i = 1 \cdots, m$ becomes a linear constraint in the $2k+d+1$ unknowns and a solution to this system can now be found by matrix inversion.

Algorithm 2.6 ($Solve(x_1, x_2, ..., x_m, s_1, s_2, ..., s_m)$)

NOTA: remember that 2k+d=m-1, thus this is a square matrix.

NOTA: remember that w_k=1.

$$\begin{pmatrix} 1 & x_1 & \dots & x_1^{k+d} & -s_1 & \dots & -s_1 x_1^{k-1} \\ 1 & x_2 & \dots & x_2^{k+d} & -s_2 & \dots & -s_2 x_2^{k-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_i & \dots & x_i^{k+d} & -s_i & \dots & -s_i x_i^{k-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & \dots & x_m^{k+d} & -s_m & \dots & -s_m x_m^{k} \end{pmatrix} \begin{pmatrix} n_0 \\ n_1 \\ \vdots \\ \vdots \\ n_{k+d} \\ w_0 \\ w_1 \\ \vdots \\ w_{k-1} \end{pmatrix} = \begin{pmatrix} s_1 x_1^k \\ s_2 x_2^k \\ \vdots \\ \vdots \\ s_i x_i^k \\ \vdots \\ \vdots \\ s_m x_m^k \end{pmatrix}$$

The Berlekamp-Welch Decoder

It may be noted that the algorithm presented here for finding W and N is not the most efficient known. Berlekamp and Welch [5] present an $O(m^2)$ algorithm for finding N and W, but proving the correctness of the algorithm is harder. The interested reader is referred to [5] for a description of the more efficient algorithm.