Error Correcting Codes: Combinatorics, Algorithms and Applications **(Fall 2007)**

Lecture 27: Berlekamp-Welch Algorithm

October 31, 2007

Lecturer: Atri Rudra *Scribe: Michel Kulhandjian*

In the last lecture, we discussed unique decoding of RS codes and briefly went through the Berlekamp-Welch algorithm. In today's lecture we will study Berlekamp-Welch algorithm in more detail.

Recall that the $[n, k, n-k+1]_q$ Reed-Solomon code regards a message as a polynomial $P(X)$ of a degree at most $k-1$, and the encoding of a message $\mathbf{m} = (m_0, \ldots, m_{k-1})$ is $(P(\alpha_1), \ldots, P(\alpha_n))$. Here, $m_i \in \mathbb{F}_q$, $k \leq n \leq q$, and $P(X) = \sum_{i=0}^{k-1} m_i X^i$. Now let us look at the decoding problem of Reed-Solomon codes. Suppose we are given distinct values $\alpha_1, \ldots, \alpha_n$ where $\alpha_i \in \mathbb{F}_q$ with received word $\mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{F}_q^n$ and parameters k and $e < \frac{n-k+1}{2}$, where e is an upper bound on the number of errors which occurred during transmission. Our goal is to find a polynomial $P(X) \in \mathbb{F}_q[X]$ of degree at most $k-1$, such that $P(\alpha_i) \neq y_i$ for at most e values of $i \in [n]$ (assuming such a $P(X)$ exists). Although this problem is quite non-trivial one, a polynomial time solution can be found for this problem. This solution dates back to 1960 when Peterson [2] came up with a decoding algorithm for the more general BCH code that runs in time $O(n^3)$. Later Berlekamp and Massey sped up this algorithm so that it runs in $O(n^2)$. There is an implementation using Fast Fourier Transform that runs in time $O(n \log n)$. We will not discuss these faster algorithms, but will study another algorithm due to Berlekamp and Welch. More precisely, we will use the Gemmell-Sudan description of the Berlekamp-Welch algorithm[1].

1 Berlekamp-Welch Algorithm

We start by describing the Berlekamp-Welch algorithm.

Input: $n \geq k \geq 1$, $0 < e < \frac{n-k+1}{2}$ and n pairs $\{(\alpha_i, y_i)\}_{i=1}^n$ with α_i distinct. Let $\mathbf{y} = (y_1, \ldots, y_n)$.

Output: Polynomial $P(X)$ of degree at most $k-1$ or fail.

Step 1: Compute a non-zero polynomial $E(X)$ of degree exactly e, and a polynomial $Q(X)$ of degree at most $e+k-1$ such that

$$y_i E(\alpha_i) = Q(\alpha_i) \quad 1 \leq i \leq n. \tag{1}$$

If such polynomials do not exist output fail.

Step 2: If $E(X)$ does not divide $Q(X)$ output fail else compute $P(X) = \frac{Q(X)}{E(X)}$. If $\Delta(\mathbf{y}, (P(\alpha_i))_i) > e$ then output fail else output $P(X)$.

Remark 1.1. *Notice that computing $E(X)$ a as hard as finding the solution polynomial $P(X)$. If one knew $E(X)$, then one can use erasure decoding for RS codes to recover $P(X)$. Also in some cases, the polynomial $Q(X)$ is as hard to find as $E(X)$. E.g., given $Q(X)$ and $\mathbf{y}$ (such that $y_i \neq 0$ for $1 \leq i \leq n$) one can find the error locations by checking positions where $Q(\alpha_i) = 0$. While each of these polynomials $e(X)$, $Q(X)$ is hard to find individually, together they are easier to find.*

Note that if the algorithm does not output fail, then the algorithm produces a correct output. Thus, to prove the correctness of the Berlekamp-Welch algorithm, we just need the following result.

Theorem 1.2. *If $(P(\alpha_i))_i$ is transmitted (where $P(X)$ is a polynomial of degree at most $k-1$) and at most e errors occur, then the Berlekamp-Welch algorithm outputs $P(X)$.*

The proof of the theorem above follows from the subsequent claims.

Claim 1.3. *There exist a pair of polynomials $E(X)$ and $Q(X)$ that satisfy* **Step 1** *such that $\frac{Q(X)}{E(X)} = P(X)$.*

Note that now it suffices to argue that $\frac{Q_1(X)}{E_1(X)} = \frac{Q_2(X)}{E_2(X)}$ for any pair of solutions that satisfy **Step 1** since Claim 1.3 above can then be used to see that ratio must be $P(X)$. Indeed, we will show this to be the case:

Claim 1.4. *If any two distinct solutions $(E_1(X), Q_1(X)) \neq (E_2(X), Q_2(X))$ satisfy* **Step 1**, *then they will satisfy*

$$\frac{Q_1(X)}{E_1(X)} = \frac{Q_2(X)}{E_2(X)}.$$

Proof of Claim 1.3. We just take $E(x)$ to be the error-locating polynomial for $P(X)$ and let $Q(X) = P(X)E(X)$ where $\deg(Q(X)) \leq \deg(P(X)) + \deg(E(X)) \leq e + k - 1$. In particular, define $E(X)$ as the following polynomial of degree exactly e:

$$E(X) = X^{e-\Delta(\mathbf{y},(P(\alpha_i))_i)} \prod_{1 \leq i \leq n | y_i \neq P(\alpha_i)} (X - \alpha_i) \tag{2}$$

By definition, $E(X)$ is a degree e polynomial with the following property:

$$E(\alpha_i) = 0 \quad \text{iff} \quad y_i \neq P(\alpha_i).$$

We now argue that $E(X)$ and $Q(X)$ satisfy (1). Note that if $E(\alpha_i) = 0$, then $Q(\alpha_i) = P(\alpha_i)E(\alpha_i) = y_i E(\alpha_i) = 0$. When $E(\alpha_i) \neq 0$, we know $P(\alpha_i) = y_i$ and so we still have $P(\alpha_i)E(\alpha_i) = y_i E(\alpha_i)$, as desired. □

Proof of Claim 1.4. Note that the degrees of the polynomials $Q_1(X)E_2(X)$ and $Q_2(X)E_1(X)$ are at most $2e + k - 1$. Let us define polynomial $R(X)$ with degree at most $2e + k - 1$ as follows:

$$R(X) = Q_1(X)E_2(X) - Q_2(X)E_1(X). \tag{3}$$

Furthermore, from **Step 1** we have, for every $i \in [n]$,

$$y_i E_1(\alpha_i) = Q_1(\alpha_i) \quad \text{and} \quad y_i E_2(\alpha_i) = Q_2(\alpha_i). \tag{4}$$

Substituting (4) into (3) we get for $1 \le i \le n$:

$$\begin{aligned} R(\alpha_i) &= (y_i E_1(\alpha_i))E_2(\alpha_i) - (y_i E_2(\alpha_i))E_1(\alpha_i) \\ &= 0. \end{aligned}$$

The polynomial $R(X)$ has n roots and

$$\begin{aligned} \deg(R(X)) &\le e + k - 1 + e \\ &= 2e + k - 1 \\ &< n, \end{aligned}$$

Where the last inequality follows from the upper bound on e. Since $\deg(R(X)) < n$, then we get that the polynomials $Q_1(X)E_2(X)$ and $Q_2(X)E_1(X)$ agree on more points than their degree, and hence they are identical. Note that as $E_1(X) \neq 0$ and $E_2(X) \neq 0$, this implies that $\frac{Q_1(X)}{E_1(X)} = \frac{Q_2(X)}{E_2(X)}$, as desired. □

1.1 Implementation of Berlekamp-Welch Algorithm

In **Step 1**, $Q(X)$ has $e + k$ unknowns and $E(X)$ has $e + 1$ unknowns. For each $1 \le i \le n$, the constraint in (1) is a linear equation in these unknowns.

Thus, we have a system of n linear equations in $2e + k + 1 < n + 2$ unknowns. By claim 1.3, these system of equations have a solution. The only extra requirement is that the degree of polynomial $E(X)$ should be exactly e. We have already shown $E(X)$ in equation (2) to satisfy this requirement. So we add a constraint that the coefficient of X^e in $E(X)$ is 1. We have $n + 1$ linear equation in at most $n + 1$ variables, which we can solve in time $O(n^3)$, e.g. by Gaussian elimination. Finally, note that **Step 2** can be implemented in time $O(n^3)$ by "long division."

Theorem 1.5. *For any $[n, k]_q$ Reed-Solomon code it can be uniquely decoded in $O(n^3)$ time up to $\frac{D}{2} = \frac{n-k+1}{2}$ number of errors.*

2 Errors And Erasures decoding of RS codes

We have seem that there exists polynomial time constructible codes that lie on the Zyablov bound and can be corrected up to $1/4$ of their designed distance. Next, we look into codes that correct up to $\frac{1}{2}$ design distance for explicit codes on the Zyablov bound. For this one we will need correction from *both* errors and erasures. Berlekamp-Welch algorithm corrects up to $e < \frac{D}{2}$ errors for $[N, K, D]_Q$ RS codes. We have already seen that we can correct $s < D$ erasures.[1] We will prove the following more general result in the next lecture:

[1]The number of unknown positions is s and the number of known position is $N - s \ge N - D + 1 = K$. Having K constrains we can recover polynomial of degree at most $K - 1$ by interpolation (K unknowns and K equations) in $O(N^3)$ time.

Proposition 2.1. *Reed-Solomon codes can be corrected from e errors and s erasures as long as $2e + S < D$ in $O(N^3)$ time.*

References

[1] P. Gemmell and M. Sudan. Highly resilient correctors for polynomials. *Information processing letters*, 43(4):169–174, September 1992.

[2] W. Wesley Peterson. Encoding and error-correction procedures for bose-chadhuri codes. *IEEE Transactions on Information Theory*, 6:459–470, 1960.