6.897 Algorithmic Introduction to Coding Theory October 16, 2001

Lecture 11

Lecturer: Madhu Sudan *Scribe: Rui Fan*

1 Overview

- Decoding RS codes.
- Abstract decoding algorithm.

2 Decoding Reed-Solomon codes

So far we've seen how to encode messages with the RS code. Recall that the RS encoding of a message $m = \langle m_0, \ldots, m_{k-1} \rangle$ is $\langle p(\alpha_1), \ldots, p(\alpha_n) \rangle$. Here, $m_i \in \mathbb{F}_q$, $k \leq n \leq q$, and $p(x) = \sum_{i=0}^{k-1} m_i x^i$. The decoding problem for RS codes can be stated as follows:

Suppose we are given distinct values $\alpha_1, \ldots, \alpha_n$, $\alpha_i \in \mathbb{F}_q$. Let $r_1, \ldots, r_n$ be our received vector, $r_i \in \mathbb{F}_q$. Let k and e be parameters, where e is $\geq$ the number of errors which occurred in the received vector. Our goal is to find a polynomial p of degree $\leq k$, such that $p(\alpha_i) \neq r_i$ for at most e values of i.

Berlekamp and Welch gave an algorithm in 1986 which finds a unique p for the above problem if p exists, and $e \leq (n-k)/2$. Interestingly, finding the actual solution is the only way we know of to prove that a solution exists.

2.1 Error-locating polynomial

The BK algorithm starts by defining an *error-locating polynomial.* This is a polynomial which is associated with the inputs, but which we don't (yet) know how to find. Nevertheless, it has some nice properties which eventually lead to a decoding algorithm.

Definition 1 *Let e, k be some parameters. Let $\alpha_1, \ldots, \alpha_n$ and $r_1, \ldots, r_n$ be such that there exists a polynomial p of degree $\leq k$ such that $p(\alpha_i) \neq r_i$ for $\leq e$ values of i. $E(x)$ is an* error-locating polynomial *for the above inputs if we have*

1. $E(\alpha_i) = 0$ *if* $p(\alpha_i) \neq r_i$.
2. $\deg(E) \leq n - k - 1$, *and* $E \neq 0$.

Note that given E, we can compute p, by finding the i's for which $E(\alpha_i) = 0$, replacing r_i by ? for those i's, and doing erasure decoding on the resulting vector of r_i's. This works because there are at most $n - k - 1 < d - 1$ i's for which $E(\alpha_i) = 0$. We now list some properties of E.

1. E is nonzero, and has small degree.
2. $E \cdot p$ is a small degree polynomial.
3. $\forall i : (r_i - p(\alpha_i))E(\alpha_i) = 0$

Property 3 holds because either $r_i = p(\alpha_i)$, or $r_i \neq p(\alpha_i)$ but $E(\alpha_i) = 0$.

Define $N(x) = E(x) \cdot p(x)$. By property 3, $N(\alpha_i) = E(\alpha_i)p(\alpha_i) = r_i E(\alpha_i)$, for all i. We're now ready for the decoding algorithm.

2.2 Berlekamp-Welch decoding algorithm

Input: $\alpha_1, \ldots, \alpha_n, r_1, \ldots, r_n$, with $\alpha_i, r_i \in \mathbb{F}_q$. k, and $e \leq (n-k)/2$. Step 1. Find polynomials $N(x) \neq 0$, $E(x) \neq 0$ such that

1. $N(\alpha_i) = r_i E(\alpha_i)$ for all i.
2. $\deg(N) \leq e + k$.
3. $\deg(E) \leq e$.

Step 2. Output $N(x)/E(x)$ if $E|N$. Otherwise output "no such polynomial".

2.2.1 Correctness

To prove this algorithm is correct, we need to prove the following 3 things:

1. There exists N, E satisfying the conditions of step 1.
2. The algorithm can be performed efficiently.
3. The solution it outputs is unique.

Proof of 1. Define $E(x) = \prod_{i:p(\alpha_i) \neq r_i}(x - \alpha_i)$. If there are $\leq e$ errors, which is the only case for which the algorithm needs to work correctly, then $\deg(E) \leq e$. Now define $N(x) = E(x)p(x)$. We saw earlier that for such an N, $N(\alpha_i) = r_i E(\alpha_i)$ for all i.

Proof of 2. We first note that we can efficiently compute the polynomial division in step 2. Step 1, finding $N(\alpha_i) = r_i E(\alpha_i)$ for all i, is solving a homogeneous linear system of n equations

$$\sum_{j=0}^{e+k} N_j \alpha_i^j = r_i \sum_{j=0}^{e} E_j \alpha_i^j$$

in the unknowns $N_0, \ldots, N_{e+k}, E_0, \ldots, E_e$.We already saw that a solution to this system exists, in the proof of property 1. The straightforward way of solving this system works in time $O(n^3)$. Welch and Berlekamp gave an iterative algorithm which makes t passes over the input, where t is the number of errors, and has a total running time of $O(t \cdot n)$. The fastest known algorithm, based on the Fast Fourier Transform, takes time $O(n\text{polylog}n)$.

Proof of 3. Assume that we have pairs (N, E), and (N', E') which both satisfy the conditions of step 1. We'll show that $\frac{N}{E} = \frac{N'}{E'}$, so that the solution output by the algorithm is unique. We'll do this by cross-multiplying and showing that $N \cdot E' = N' \cdot E$. We consider 2 cases.

Case 1. $r_i = 0$: Then $N(\alpha_i) = N'(\alpha_i) = 0$, so we're done.

Case 2. $r_i \neq 0$. Then $N(\alpha_i)N'(\alpha_i) = r_i E(\alpha_i)N'(\alpha_i) = N(\alpha_i) r_i E'(\alpha_i)$. Dividing through by r_i, we get that $E(\alpha_i)N'(\alpha_i) = N(\alpha_i)E'(\alpha_i) \forall i \in [n]$. But since $N \cdot E'$ and $N' \cdot E$ have degree $\leq 2e + k$, while $n > 2e + k$, this implies that $N \cdot E' = N' \cdot E$ as polynomials. Thus, $\frac{N}{E} = \frac{N'}{E'}$, and the output of the algorithm is unique.

3 Abstract decoding procedure

It turns out that the BK algorithm represents a style of decoding which is not unique to RS codes. We now describe some work by Pellikaan, Kötter and Duursma which abstracts the BK algorithm, so that it can be used to decode other codes, such as algebraic-geometry codes. First we give some definitions.

Definition 2 *Let $u, v \in \mathbb{F}_q^n$, and $A, B \subseteq \mathbb{F}_q^n$. Define $u \star v = (u_1 v_1, \ldots, u_n v_n)$, and define $A \star B = \{a \star b \mid a \in A, b \in B\}$.*

The idea of the abstract decoding procedure is that given an e-error correcting code C which we want to decode, we construct an *error-locator code* E, such that $E \star C$ is contained in some "nice" linear code N. By "nice", we mean that N has large distance. Specifically, we want codes E and N to have the following properties:

1. $\dim(E) > e$.
2. $E \star C \subseteq N$.
3. $\mathrm{dist}(N) > e$.
4. $\mathrm{dist}(N) > n - \mathrm{dist}(E)$.

Then, given a received vector $(r_1, \ldots, r_n)$, such that there exists a $c \in C$, $c = (c_1, \ldots, c_n)$ with $\Delta(r, c) \leq e$, we want an algorithm to find c.

3.1 Abstract decoding algorithm

The abstract decoding algorithm parallels the concrete BK decoding algorithm for RS codes. This algorithm works correctly if there are $\leq e$ errors in r.

Input: A vector $r \in \mathbb{F}_q^n$ which we want to decode to a codeword in C. Also, codes E and N with the properties described in the last section.

Step 1. Find $a \in E$ and $b \in N$, $(a, b) \neq (0, 0)$, such that $a \star r = b$, and $a_i = 0$ if $r_i \neq c_i$.

Step 2. For any i with $a_i = 0$, set $r_i =?$. Then do erasure decoding on the resulting vector r to find c. If this does not result in a codeword, output "no such codeword".

3.1.1 Correctness

As in the proof of the BK decoding algorithm, we need to argue 3 conditions.

1. A solution (a, b) exists.
2. The algorithm is efficient.
3. The output c is unique.

Proof of 1. We first show that there exists $a \in E$, $a \neq 0$, such that $a_i = 0$ if $r_i \neq c_i$. We assume that $\leq e$ errors occurred, since the algorithm only needs to work corectly in this case. Then $a_i = 0$ for at most e values of i. This places at most e linear constraints on a. But since $\dim(E) > e$, we know that there is a nonzero vector $a \in E$ satisfying these constraints. Now, define $b = a \star c$. Clearly $b \in N$. We also have $b_i = a_i r_i$. This is because either we have $r_i = c_i$, in which case $b_i = a_i c_i = a_i r_i$, or, we have $r_i \neq c_i$, but then $a_i = 0$ and $a_i c_i = a_i r_i = 0$. Thus, there does exist $(a, b) \neq (0, 0)$ with the properties required by step 1.

Proof of 2. To see that step 1 can be performed efficiently, note that E and N are both linear spaces. Thus the requirement that $a \in E$, $b \in N$ are linear constraints. Also, the requirement that $a \star r = b$ is linear, since it just means $a_i r_i = b_i$. Therefore, finding the required a and b is solving a homogeneous linear system, which can be done efficiently. Also, step 2 can be done efficiently, since we saw that doing erasure decoding is also solving a linear system. Thus, the algorithm is efficient.

Proof of 3. We show that the output c is unique with 2 lemmas.

Lemma 1 For any (a, b) satisfying the conditions of step 1, we hav $a \star c = b$.

Proof We know that $a \star r = b$. Suppose $a \star c = b'$. We show that $b = b'$. Since $b'_i = a_i c_i$, and $b_i = a_i r_i$, we have $b_i \neq b'_i$ only if $c_i \neq r_i$. Since there are $\leq e$ errors, there are at most e such indices, so $\Delta(b, b') \leq e$. But $b, b' \in N$, and $\mathrm{dist}(N) > e$. Thus, $b = b'$.

Lemma 2 There is a unique c such that $a \star c = b$.

Proof Suppose that $a \star c' = b$. We need to show that $c' = c$. We have $a \star c = a \star c'$. Also, since $a \in E$, $a_i \neq 0$ for at least $\mathrm{dist}(E)$ i's. This means c and c' agree on at least $\mathrm{dist}(E)$ coordinates, and so $\Delta(c, c') < n - \mathrm{dist}(E)$. But $c, c' \in C$, and $\mathrm{dist}(C) > n - \mathrm{dist}(E)$. Thus, $c = c'$.

Thus, the abstract decoding algorithm outputs a unique c. It's not easy to find E and N which satisfy the requirements of the abstract decoding algorithm. But such codes do exist, and next time, we'll see an application of the abstract decoding algorithm.

4 References

Matt Lepinski's notes for lecture 11 of the 2001 version of this course.