6.897 Algorithmic Introduction to Coding Theory October 21, 2002

Lecture 12

Lecturer: Madhu Sudan *Scribe: David Woodruff*

1 Overview

- Decoding with error-locating pairs
- Decoding concatenated codes

2 Decoding with Error-locating Pairs

We recap our analysis of e-error locating pairs from last time. For more detail, refer to Lecture 11. We defined $u \star v = (u_1 v_1, \ldots, u_n v_n)$ for $u, v \in \mathbb{F}_q^n$, and $A \star C = \{a \star b | a \in A, b \in B\}$ for $A, B \subseteq \mathbb{F}_q^n$. Recall that (A, B) is an e-error locating pair for a linear code C iff

1. A, B are linear codes
2. $A \star C \subseteq B$
3. $\dim(A) > e$
4. $\Delta(B) > e$
5. $\Delta(C) > n - \Delta(A)$

Finding an (A, B) pair which satisfies properties 2-4 is nontrivial. We will touch more on this when we talk about algebraic geometry codes later in the lecture.

2.1 Abstract Decoding Algorithm

Last time we presented a simple decoding algorithm for C given an e-error locating pair (A, B) for C. Note that the algorithm is nonuniform since it knows A and B. Also it is only required to work correctly if there are $\leq e$ errors in the received vector $r = (r_1, \ldots, r_n)$. The algorithm is:

Input: Received vector r, A, B

1. Find $a \in A$, $b \in B$, $(a, b) \neq (0, 0)$, such that $a \star r = b$ and $a_i = 0$ if $r_i \neq c_i$.
2. Use erasure decoding to find $c \in C$ such that $c_i = r_i$ if $a_i \neq 0$, and output c.

2.1.1 Correctness

The following were some of the key properties in showing correctness (see Lecture 11 for the proofs):

1. Show that (a, b) as required in Step 1 exist: Use $\dim(A) > e$ to prove there exists a nonzero $a \in A$ that is zero if $r_i \neq c_i$. Let a be any such vector. Let $b = a \star c$. It then follows that $a \star r = b$.
2. Show that any pair (a', b') found in Step 1 satisfies $a' \star c = b'$: Suppose to the contrary, $a' \star c = b''$ for such a pair (a', b') found in Step 1, so that $a' \star r = b'$. Then b' and b'' agree whenever $r_i = c_i$, which means they disagree on at most e locations. But since $\Delta(B) > e$, we have that $b' = b''$.
3. Show there is a unique c such that $a \star c = b$: Suppose to the contrary, $a \star c = a \star c'$. But $a_i \neq 0$ for at least $\Delta(A)$ coordinates, and hence c and c' agree on these coordinates, and so $\Delta(c, c') < n - \Delta(A)$, a contradiction.

2.2 Typical Cases for Error-locating Pairs

How nontrivial is it to find an e-error-locating pair (A, B) for an $[n, k, d]_q$ code C? Suppose A is an $[n, k_1, ?]_q$ code, i.e., a linear code with dimension k_1 and unspecified minimum distance. Then the dimension of $A \star C$ is at most $k \cdot k_1$, since if $\{a_1, \ldots, a_{k_1}\}$ is a basis for A, and if $\{c_1, \ldots, c_k\}$ is a basis for C, then $\{a_i c_j, 1 \leq i \leq k_1, 1 \leq j \leq k\}$ spans $A \star C$. For an A chosen at random you expect the dimension of $A \star C$ to be exactly $k \cdot k_1$. From the Singleton bound, $\Delta(B) \leq n - \dim(B) + 1 = n - \dim(A \star C) + 1 \approx n - kk_1 + 1$. This hurts us since kk_1 is large, making it difficult to achieve property 4 of an e-error locating pair.

Lets look at RS codes and why we were able to find error-locating pairs for them. Recall that A was contained in the vector space of polynomials of degree at most k_1 and C was contained in the vector space of polynomials of degree k. What's nice about polynomial vector spaces is that their product, $A \star C$ is contained in the vector space of polynomials of degree $k + k_1$. Hence, $\dim(A \star C) = \dim(A) + \dim(C) = k + k_1$ for RS decoding. This dimension is much smaller than kk_1, which was what was achieved for random A.

2.3 Applications to Algebraic Geometry Codes

Although the above analysis is discouraging, there are some codes other than RS codes for which one can construct error-locating pairs. One such application is algebraic geometry codes. Recall that in these codes we fix some subset $S \subseteq \mathbb{F}_q^n$ with $|S| = n$. The messages are then m-variate polynomials, and to encode we evaluate such a polynomial on the points in S. We were able to achieve good distance in part because of the following properties of the order function ord on polynomials. We will omit the definition of ord, instead concentrating on its properties that allow application of our abstract decoding algorithm. We have:

1. $ord(f + g) \leq \max(ord(f), ord(g))$.
2. $ord(f \star g) = ord(f) + ord(g)$, except if $f, g = 0$.
3. $\exists$ a function of all orders with a finite set of exceptions. The cardinality of this set is referred to as the *genus* of S. From algebraic geometry, we know there exists an S with $|S| = n = q^{\frac{m}{2}}$ and $genus(S) \leq \frac{n}{\sqrt{q}-1}$ (Hurwitz's genus formula).
4. if $ord(f) = k$, then f has at most k zeros in S.

Properties 1 and 2 are quite similar to the properties of order on polynomials. It is in fact property 4 which when combined with the other properties allows us to use our abstract decoding algorithm on these codes. This property is quite similar to the property for degree on univariate polynomials and, using the notation from the previous section, it is what ensures $\dim(A \star C) = k + k_1$ instead of $k \cdot k_1$. Note, the reader is referred to Matt Lepinski's Lecture 11 notes from last year for a more rigorous treatment of algebraic geometry codes in the context of our abstract decoding algorithm.

2.4 Chinese Remainder Codes

In the Chinese Remainder Codes the message space is the set of integers from 0 to $K - 1$ (i.e., Z_K) where $K = p_1 \cdots, p_k$ is the product of distinct primes $p_1 < p_2 < \cdots < p_k$. For $m \in Z_K$, we encode it as $([m]_{p_1}, \ldots, [m]_{p_k})$, where $[m]_{p_i} = m \bmod p_i$. From the Chinese Remainder Theorem (CRT), we know that given values for $[m]_{p_1}, \ldots, [m]_{p_k}$, we can reconstruct m. Note that this code is not linear.

What if instead of encoding m as $([m]_{p_1}, \ldots, [m]_{p_k})$ we were to encode m as $([m]_{p_1}, \ldots, [m]_{p_n})$ for $n \geq k$ distinct primes $p_1 < \cdots < p_n$? From CRT, we know that we can reconstruct m from any k of these n residues. Hence, these codes can correct up to $n - k$ erasures, and by the Singleton bound, this implies they have distance $n - k + 1$. One can now ask whether or not it's possible to correct $\frac{(n-k)}{2}$ errors.

It turns out that there is a decoding algorithm similar in spirit to our abstract decoding algorithm that was presented in [1] by Goldreich, Ron, and Sudan.

3 Decoding concatenated codes

We now shift our attention to decoding concatenated codes, specifically Forney codes. Recall the definition of a concatenated code. We were given an $[N, K, D]_Q$ *outer code* C_1 with encoding function E_1 and an $[n, k, d]_q$ *inner code* C_2 with encoding function E_2. We impose the constraint $Q = q^k$. Concatenated codes can be looked at as applying E_1 to a message m, obtaining a codeword $u \in \mathbb{F}_Q^N$. One can then interpret u as a vector $(u_1, \ldots, u_N) \in \mathbb{F}_{q^k}^N$ by splitting it coordinate-wise into individual elements $u_i \in \mathbb{F}_Q \equiv \mathbb{F}_{q^k}$. Now one applies E_2 to each u_i to get a vector $v = (v_1, \ldots, v_n)$, with $E_2(u_i) = v_i$. Using the right representation, one can ensure that the concatenation of linear codes is still a linear code. The resulting code is thus an $[Nn, Kk, Dd]_q$ code. Showing the minimum distances multiply was done in Lecture 6.

Suppose $(v_1, \ldots, v_n)$ is transmitted across a noisy channel and $(r_1, \ldots, r_n)$ is the received vector. To decode we can imagine first decoding the the inner code to obtain a vector $(y_1, \ldots, y_n)$ and then interpreting the vector as an element in $\mathbb{F}_Q^N$ and then decoding the outer code. In the codes we have considered so far, the outer code is usually some well-studied code like the RS code for which we can apply efficient decoding algorithms. The inner code, although less understood, is a very small code so we can just use brute force to decode, enumerating all possible messages y_i, and finding $\text{argmin}_{y_i} \Delta(E_2(y_i), r_i)$. Since the inner code is small, the time this takes is polynomial in the length of the concatenated code, and hence so is the entire decoding algorithm.

Unfortunately the number of errors this decoding algorithm can correct is $< \frac{dD}{4}$. This is because to fail in the outer decoding step, we only need $\frac{D}{2}$ i's such that $y_i \neq u_i$. Then, for each location i, we only need $\frac{d}{2}$ errors in the transmission of v_i so that $y_i \neq u_i$. Thus a total of $\frac{dD}{4}$ errors may lead to a decoding failure. One can reverse this argument to show the above decoding algorithm does in fact correct at least $(d-1)(D-1)$ errors.

We now restrict our attention to Forney codes. Forney was able to improve upon this decoding algorithm by making the observation that the outer code can benefit from correcting both erasures and errors instead of only errors.

Proposition 1 *Suppose C_1 is an $[N, N-D+1, D]_Q$ RS code. Suppose $r \in (\mathbb{F}_Q \cup \{?\})^N$ is a vector derived from a codeword $c \in C_1$ with e errors and s erasures. Then c can be recovered efficiently given r, e, and s as long as $2e + s < D$.*

Proof Removing s coordinates of C results in an $[N-s, N-D+1, D-s]$ RS code, from which $\frac{D-s-1}{2}$ errors can be corrected efficiently. The proposition follows. ■

As we brute force decode r_i into y_i, if there are too many errors in r_i, we will just replace the ith coordinate with ? instead of correcting it. At what point are there too many errors? Certainly if there are more than $\frac{d}{2}$ errors we should erase. Forney devised a probabilistic decoding algorithm for the case when the number of errors between r_i and the encoding of the nearest codeword y_i, $\Delta(r_i, E_2(y_i))$, is e_i with $0 \leq e_i \leq \frac{d}{2}$. We will denote the actual number of errors between r_i and v_i as e_i'. As a minor note, for most codes, the nearest codeword c to a received vector r can in fact be such that $\Delta(c, r) > \frac{d}{2}$, although for the Hamming code this was not the case.

The randomized algorithm of Forney is:

Input: Received vector r

1. For $1 \leq i \leq n$, brute-force decode r_i into y_i, i.e., find y_i such that $\Delta(r_i, E_2(y_i))$ is as small as possible for all i.
2. For each i, set $y_i =?$ with probability $\min(1, \frac{2e_i}{d})$.
3. Perform an error and erasure decoding algorithm on $(y_1, \ldots y_n)$ to obtain and output m' (note that the algorithm is only required to be correct for # errors $= \Sigma e_i' < \frac{Dd}{2}$).

The above algorithm is actually linear time in the length of the concatenated code NnQ and in the time taken to perform error and erasure decoding. We propose the following:

Proposition 2 *Let E_i be an indicator variable which is 1 iff you get an error in the ith coordinate. Let S_i be an indicator variable which is 1 iff you get an erasure in the ith coordinate. Let $E = \Sigma_i E_i$ and $S = \Sigma_i S_i$. If $\Sigma_i e_i' < \frac{Dd}{2}$, then $\boldsymbol{E}[2E + S] < D$, where $\boldsymbol{E}[X]$ denotes random variable X's expectation.*

Proof By linearity of expectations, it suffices to prove $\mathbf{E}[2E_i + S_i] < \frac{e_i'}{\frac{d}{2}}$. By definition, $\mathbf{E}[S_i] = \frac{e_i}{\frac{d}{2}}$. Hence it suffices to prove $\mathbf{E}[E_i] \leq \left(\frac{e_i' - e_i}{\frac{d}{2}}\right)\frac{1}{2} = \frac{e_i' - e_i}{d}$. Now if $y_i = u_i$, then $\mathbf{Pr}[\text{error in position } i] = 0 = \mathbf{E}[E_i] \leq \frac{e_i' - e_i}{d}$ since $e_i' \geq e_i$. On the other hand, if $y_i \neq u_i$, then $e_i' = \Delta(v_i, r_i) \geq \Delta(v_i, E_2(y_i)) - \Delta(r_i, E_2(y_i)) \geq d - e_i$. Hence in this case,

$$\mathbf{E}[E_i] = 1 - \frac{e_i}{\frac{d}{2}} = \frac{d - 2e_i}{d} = \frac{d - e_i - e_i}{d} \leq \frac{e_i' - e_i}{d},$$

concluding the proof. ■

Note that we have just shown the desired result in expectation. Rather than achieving a high probability result, we can instead derandomize the above algorithm. The algorithm is trivial to derandomize since linearity of expectations holds regardless of any dependencies amongst the E_i, S_i. Hence, to derandomize, we choose a threshold τ and in Step 2 we simply set $y_i =?$ if $\frac{2e_i}{d} > \tau$. By the probabilistic method we know there exists a τ such that $\mathbf{E}[2E + S] < D$. Note that we can just choose τ in the set $\{0, 1\} \cup \{\frac{2e_i}{d} | 1 \leq i \leq N\}$. This is because for any other value of τ, our algorithm will output a vector m' which is the same for some $\tau \in \{0, 1\} \cup \{\frac{2e_i}{d} | 1 \leq i \leq N\}$

Hence, we have a deterministic decoding algorithm for Forney codes that runs in polynomial time, solving the unambiguous decoding problem. Note that Forney was not working in the adversarial model. Instead his goal was to correct errors when every transmitted bit is flipped with some probability p. Working in the Shannon model, he needed a code of rate $1 - H(p) - \epsilon$ and a decoding error probability of $2^{-\epsilon^2 n}$. With an inner code $[n, n(1 - H(p) - \frac{\epsilon}{2}), (p + \frac{\epsilon}{10})]$ of rate $1 - H(p) - \frac{\epsilon}{2}$, and an RS outer code $[N, N(1 - \frac{\epsilon}{2}), \frac{\epsilon N}{2}]$, one gets a terrible relative minimum distance. Forney devised a scheme which achieves polynomial time encoding, polynomial time decoding, and rate arbitrarily close to the channel capacity in the Shannon model using concatenated codes.

4 References

- Rui Fan's notes for Lecture 11 of this year's version of the course.
- Matt Lepinski's notes for Lecture 11 of the 2001 version of this course.

[1] Oded Goldreich, Dana Ron, and Madhu Sudan. Chinese remaindering with errors. *IEEE Transactions on Information Theory*, 46(5):1330-1338, July 2000. Extended version appears as ECCC Technical Report TR98-062 (Revision 4), http://www.eccc.uni-trier.de/eccc