CSE522/Math581EB Coding Theory **Fall 2001**

Lecture 6: October 25

Lecturer: Elchanan Mossel *Scribe: Rebecca Weinhold*

Last time, we found practical methods for constructing codes in polynomial time, but we didn't address the problems of encoding and decoding. We would like to encode and decode in reasonable time.

6.1 Encoding, Decoding, and Correcting Erasures

6.1.1 Encoding

Given an $[n, k, d]_q$ code, encoding is done by mapping $x \in \mathcal{F}_q^k$ to $xG \in M_{n\times n}(\mathcal{F}_q)$; this takes $O_q(n \cdot k)$ steps (O_q counts arithmetic operations over the field $\mathcal{F}_q$). In the future we will try to find codes for which the encoding could be done in linear time.

6.1.2 Decoding

Finding the time necessary for decoding depends on what we mean by "decode." The weakest and strongest definitions of decoding are:

- Weakest: If there are no errors, we solve the equation $xG = y$. This is a system of n equations in k variables, so this can be done in $O_q(n^3)$ steps. Actually, we can choose a subset of k independent equations, and then it takes $O_q(k^3)$ steps. In fact there are ways of solving this system in $O_q(k^{2.3\cdot\cdot})$ but we don't care so much about that.
- Given y, output the x such that $\Delta(xG, y)$ is minimal. The general instance of this problem for linear codes is NP-hard [BmCvT].

6.1.3 Correcting Erasures

Here we have received $y \in (\mathcal{F}_q \cup \{?\})^n$, and want to find the correct codeword x. This amounts to solving the matrix equation $xG = y$, but we delete from consideration those equations of the form $\sum x_i g_{ij} =?$. If s is the number of erasures (positions containing "?") in y, we are left with $n - s$ equations.

- If $s < d$, then this is sufficient to find a unique solution x.
- If $s \geq d$, then we can find a linear subspace of all possible solutions x.

6.1.4 Correcting errors for Reed-Solomon codes

Since correcting errors for general codes is hard, we consider the problem for specific codes. The single most important family of codes are Reed-Solomon codes. For these codes it is possible to correct up to $d/2$ errors in polynomial time! [BkWe].

Theorem 6.1 *The Reed-Solomon code $RS_{n,k,q}$ is an $[n, k, n-k+1]_q$ code for which we can decode $t \leq \frac{d-1}{2} = \frac{n-k}{2}$ errors (in $O(n^3)$ steps).*

Before proving this, we need a definition and a pair of lemmas.

Definition 6.2 *Given $y = (y_1, \ldots, y_n) \in \mathcal{F}_q^n$, E is an error-locating polynomial for p if $E(x_i) = 0$ whenever $p(x_i) \neq y_i$.*

Lemma 6.3 *Given $y = (y_1, \ldots, y_n) \in \mathcal{F}_q^n$ and $p \in \mathcal{F}_q[x]$ with $\deg p \leq k-1$ and $p(x_i) \neq y_i$ for at most t i's, there exist polynomials E and N such that*

(1) $\deg E \leq t$,

(2) $\deg N \leq t + k - 1$, *and*

(3) $N(x_i) = y_i E(x_i)$.

Moreover, we can find (N, E) in $O(n^3)$ steps.

Proof: Existence: To find an error-locating polynomial E for p, we define $E(x) = \prod_{i:y_i \neq p(x_i)} (x - x_i)$. Then defining $N(x) = p(x)E(x)$ yields the pair (N, E) where the degree of E is the number of errors. The degree of N is at most the sum of the degrees of E and p, so $\deg E \leq t$ and $\deg N \leq t + k - 1$. The pair (N, E) constructed in this way satisfies (1) and (2) in the statement of the lemma. To satisfy property (3), we need to solve the system of linear equations

$$N(x_i) = y_i E(x_i) \Rightarrow \sum_{j=0}^{t+k-1} n_j x_i^j = y_i \sum_{j=0}^{t} e_j x_i^j,$$

where we write $E(x) = \sum_{j=0}^{t} e_j x^j$ and $N(x) = \sum_{j=0}^{t+k-1} n_j x^j$. We know the x_i's and y_i's and therefore the equations are linear in the unknowns e_i and n_i. We have just seen that there exists a solution to this system. Hence we can find (N, E) that satisfy (1) through (3); doing so can be done in $O(n^3)$ time, since it is a matter of solving the system of linear equations above. ■

Lemma 6.4 *If (N, E) and (N', E') satisfy (1) through (3) in Lemma 6.3, then $\frac{N}{E} = \frac{N'}{E'}$.*

Proof: It is sufficient to show that for (N, E) and (N', E') satisfying (1) - (3), we have $NE' = N'E$. First we note that $\deg(NE') \leq \deg N + \deg E' \leq t + k - 1 + t - 2t + k - 1 \leq n - 1$, and the same holds for $\deg(N'E)$. Now using property (3), we have

$$N'E(x_i) = N'(x_i)E(x_i) = y_i E'(x_i)E(x_i) = y_i E(x_i)E'(x_i) = N(x_i)E'(x_i).$$

$N'E$ and NE' are two polynomials of degree at most $n-1$ which agree at n distinct point. Therefore $N'E = NE'$ as needed. ■

Proof:[Theorem 6.1] Given $y = (y_1, \ldots, y_n) \in \mathcal{F}_q^n$, we need to find $p \in \mathcal{F}_q[x]$ with $\deg p \leq k-1$ such that $p(x_i) \neq y_i$ for at most t positions. We first find (N, E) as in the Lemma 6.3. By Lemma 6.4 the ratio $\frac{N}{E}$ is determined uniquely. Moreover, we know that $E = \prod_{i:p(x_i) \neq y_i} (x - x_i)$, and $N(x) = p(x)E(x)$ satisfy the conditions of Lemma 6.4. Thus in $O(n^3)$ steps we have the polynomial p we seek. ■

6.1.5 Decoding Concatenated Codes

Consider a concatenated code $[n_1, k_1, d_1]_{q^{k_2}} \circ [n_2, k_2, d_2]_q$.

Claim 6.5 *If the outer code corrects up to e_1 errors in time T, then the concatenation corrects up to $\frac{e_1 d_2}{2}$ errors in time $T + O(nq^{k_2})$.*

Proof: The inner code decodes by brute force in $O(nq^{k_2})$ operations. If the number of errors at the outset is less than $\frac{d_2}{2} e_1$, then after this first decoding stage there number of errors is less than e_1. Therefore at the second stage we decode the original message. ■

6.2 Bounds

We think about bounds we have seen in asymptotic form, with rate $R = \frac{k}{n}$ and distance $\delta = \frac{d}{n}$, where $\delta \leq \frac{1}{2}$.

- Singleton bound: $1 + k + d \leq n \Rightarrow R + \delta \leq 1$.
- Hamming bound: $2^k \text{Vol}(n, \frac{d-1}{2}) \leq 2^n \Rightarrow 2^k \cdot 2^{H(\frac{d-1}{2n})n} \leq 2^n \Rightarrow k + H(\frac{d}{2n})n \leq n \Rightarrow R + H(\frac{\delta}{2}) \leq 1$. The Hamming bound is always better asymptotically than the singleton bound.
- Gilbert-Varshamov bound: $R + H(\delta) \leq 1$ implies codes exist.

If we let $\delta = 0$, then these bounds imply $R = 1$; this is not very exciting. If we let $\delta = \frac{1}{2}$, then the Gilbert-Varshamov bound yields $R = 0$, while Hamming allows $R \neq 0$. We want to resolve this discrepancy and see where does the truth lie between $R + H(\delta) \leq 1$ and $R + H(\delta/2) \leq 1$. We start by considering the Plotkin bound.

6.2.1 Plotkin bound

Notation: for $x \in \mathcal{F}_2^n$, denote by $\bar{x} \in \mathbf{R}^n$ the vector $\bar{x}_i = \begin{cases} 1 & \text{if} x_i = 0, \\ -1 & \text{if} x_i = 1. \end{cases}$

Observe that this yields $\|\bar{x}\|^2 = n$ and $\langle \bar{x}, \bar{y} \rangle = n - 2\Delta(x, y)$.

Lemma 6.6 *If $z_1, \ldots, z_m \in \mathbf{R}^n \setminus \{0\}$ such that for all $i \neq j$, $\langle z_i, z_j \rangle \leq 0$, then $m \leq 2n$.*

Proof: We prove this by induction on n. For the base case $n = 1$, the constraint $\langle z_i, z_j \rangle \leq 0$ implies that we need exactly one vector with a positive entry and exactly one vector with a negative entry. Hence $m \leq 2 = 2n$.

For the inductive step, we may assume without loss of generality that $z_1 = (1, 0, \ldots, 0)$ (there always exits an orthonormal map that maps the first vector to $(1, 0, \ldots, 0)$). For $i > 1$, $z_i = (\alpha_i, v_i)$, where v_i is a vector of the last $n - 1$ coordinates of z_i. Then since $\langle z_1, z_i \rangle \leq 0$ for all $i > 1$, we know $\alpha_i \leq 0$ for all $i > 1$. Then $\alpha_i \alpha_j \geq 0$ for all $i, j > 1$, so $0 \geq \langle z_i, z_j \rangle = \alpha_i \alpha_j + \langle v_i, v_j \rangle \Rightarrow \langle v_i, v_j \rangle \leq 0$.

At least $m-2$ of the v_i's are nonzero; otherwise if $v_1 = (1, 0, \ldots, 0)$, $v_2 = (\alpha_2, 0, \ldots, 0)$, and $v_3 = (\alpha_3, 0, \ldots, 0)$ are all in the set, then $\langle 1, \alpha_2 \rangle \leq 0$, $\langle 1, \alpha_3 \rangle \leq 0$ and $\langle \alpha_2, \alpha_3 \rangle \leq 0$, a contradiction of the base case $n = 1$. Thus the claim holds for all n. ■

Theorem 6.7 *Plotkin Bound [Plotkin]*

1. *(1) If an* $(n, k, d)_2$ *code satisfies* $d \geq \frac{n}{2}$*, then* $k \leq \log_2 2n$*. (Hadamard codes achieve this bound.)*
2. *(2) Any* $(n, k, d)_2$ *code satisfies* $k \leq n - 2d + \log_2(4d)$*.*

Proof: (1) Let C be an $(n, k, d)_2$ code with $d \geq \frac{n}{2}$, and let $x_1, \ldots, x_m$ be the codewords in C. Then $\langle \bar{x_i}, \bar{x_j} \rangle = n - 2\Delta(x_i, x_j) \leq 0$. By the previous lemma, $m \leq 2n$, so $k \leq \log_2 2n$.

(2) Let C be any $(n, k, d)_2$ code. Then $n \geq 2d$, so write $n = 2d + l$. If $l \geq k$, then the bound follows immediately. Otherwise, let $\tilde{x}$ be the most frequent l-prefix (first l positions of a word) in the code C. Now define $\tilde{C} = \{y : \tilde{x}y \in C\}$. $\tilde{C}$ is a $[2d, k - l, d]$ code that satisfies part (1), so

$$k - l \leq \log_2(4d) \Rightarrow k \leq l - \log_2(4d) = n - 2l + \log_2(4d).$$

■

References

[1] Berlekamp E. R., McEliece R. J. and vanTilborg C. A. (1978) On the inherent intractability of certain error correcting codes, *IEEE Trans. Inf. Theory*, **IT-24**,384–386.

[2] Plotkin M. (1960) Binary codes with specified minimum distance, *IRE Trans.*, **IT-6**,445–450.

[3] Berlekamp E. R and Welch L. R. (1986) Error correcting of algebraic block codes, in *US patents.*