CS395T Pseudorandomness and Combinatorial Constructions
Lecture 10, February 15, 2001
Lecturer: Prof. David Zuckerman, Scribe: Vladimir Trifonov

Let C be an $[n,k,d]_q$ code. We know that for given $\varepsilon > 0$ a binary code with $d \geq (1/2-\varepsilon)n$ and constant rate (depending on ε) can be obtained using an ε-biased distribution. Assume that $d = (1/2-\varepsilon)n$. Using unique decoding we can decode up to $d/2 = (1/4-\varepsilon/2)n$ errors. We will show a decoding technique, list decoding, which will be able to decode more errors.

Let Δ denote the Hamming distance.

Definition 1 (List-Decoding) *Let C be an $[n,k,d]_q$ code. A list-decoding procedure for C is a function which for given $u \in \mathbb{F}_q^n$ and $e > 0$ outputs all $v \in C$ such that $\Delta(v,u) \leq e$.*

1 List-Decoding of Binary Codes

Before giving an explicit list-decoding algorithm for Reed-Solomon codes, let us see an upper bound on the number of codewords of a binary code within distance $(1/2-\sqrt{\varepsilon})n$ of a given word.

We will start with a generalization of the well known mean inequalities.

Lemma 1 (Power Mean Inequality) *Let $n > 0$ and $a_1, \ldots, a_n \geq 0$. Then*

$$f(t) = \left(\frac{a_1^t + \cdots + a_n^t}{n}\right)^{\frac{1}{t}}$$

is non-decreasing. $f(t)$ is strictly increasing if not all of the a_i's are equal.

Notice that the arithmetic mean is obtained for $t = 1$, the harmonic mean for $t = -1$ and the geometric mean for $t \to 0$.

Theorem 1 (Johnson's Bound) *If C is an $[n,k,d]_2$ code with $d \geq (1/2-\varepsilon)n$, then*

$$\forall u \in \mathbb{F}_q^n \; |\{v \in C : \Delta(u,v) \leq (1/2-\sqrt{\varepsilon})n\}| \leq 1/\varepsilon.$$

Proof: Assume w.l.o.g. that $u = 0^n$ (otherwise translate the code by u). Let $\{v_1, \ldots, v_m\} \subseteq C$ be such that $w_i = wt(v_i) \leq e = (1/2-\sqrt{\varepsilon})n$. We will obtain an upper bound on m.

Let $S = \sum_{i,j=1}^m v_i \cdot v_j$, where the inner product is in $\mathbb{R}$. Let $\bar{w} = (\sum_{i=1}^m w_i)/m$. Then

$$\begin{aligned} S &= \sum_{i=1}^m w_i + \sum_{i=1}^m \sum_{j \neq i} v_i \cdot v_j \\ &\leq \sum_{i=1}^m w_i + \sum_{i=1}^m \sum_{j \neq i} \frac{w_i + w_j - d}{2} \\ &= m\bar{w} + m(m-1)\left(\bar{w} - \frac{d}{2}\right). \end{aligned}$$

The inequality is true because $w_i + w_j = \Delta(v_i, v_j) + 2v_i \cdot v_j \geq d + 2v_i \cdot v_j$.

Let $k_j = |\{i : v_{ij} = 1\}|$ (visualize a matrix with rows the vectors $\{v_1, \dots, v_m\}$ and count the number of 1's in the j-the column), for $j \in [n]$. Then $S = \sum_{j=1}^n k_j^2$. On the other hand $\sum_{j=1}^n k_j = \sum_{i=1}^m w_i = m\bar{w}$. By the Power Mean Inequality with $t = 1$ and $t = 2$,

$$\frac{\sum_{j=1}^n k_j}{n} \le \sqrt{\frac{\sum_{j=1}^n k_j^2}{n}}.$$

Therefore $S \ge (m\bar{w})^2/n$.

Combining the upper and the lower bound for S we obtain that

$$\frac{(m\bar{w})^2}{n} \le m\bar{w} + m(m-1)\left(\bar{w} - \frac{d}{2}\right)$$

Hence

$$\begin{aligned} m &\le \frac{dn}{dn + 2\bar{w}^2 - 2n\bar{w}} \\ &\le \frac{n^2}{(1/2 - \varepsilon)n^2 - 2\bar{w}(n - \bar{w})} \\ &\le \frac{1}{1/2 - \varepsilon - 2(1/2 - \sqrt{\varepsilon})(1/2 + \sqrt{\varepsilon})} \\ &= \frac{1}{\varepsilon}. \end{aligned}$$

In the second inequality we use that $(1/2 - \varepsilon)n \le d \le n$; in the third we use that $\bar{w}(n - \bar{w})$ is increasing for $\bar{w} \in [0, (1/2 - \sqrt{\varepsilon})n] \subseteq [0, n/2]$.

□

2 Algorithm for List-Decoding of Reed-Solomon Codes

Given an $[n, k, d]_q$ Reed-Solomon code C we will show a polynomial time list-decoding algorithm for up to $e \le n - \sqrt{2kn}$ errors. Define $a = n - e$ to be the agreement of C. In this notation we will show a list-decoding algorithm for agreement at least $\sqrt{2kn}$. Although this is quite a remarkable result, a random code needs agreement $\log|C| = k \log q$ ($=k \log n$ for Reed-Solomon codes), so there is still a large gap to be covered.

Definition 2 *The (a, b)-weighted degree of $x^i y^j$ is $ai+bj$. The (a, b)-weighted degree of a polynomial $Q(x, y)$ is the maximum of the (a, b)-weighted degrees of all monomials of Q.*

Algorithm (Sudan)

Input $(\{(x_i, y_i)\}_{i\in[n]}, a, k)$ (the x_i's are not necessarily distinct).

Output List of all polynomials p of degree at most k such that $p(x_i) = y_i$ for at least a values of $i \in [n]$.

1. Find $Q \not\equiv 0$ such that its $(1, k)$-weighted degree is less than a ($= \sqrt{2kn}$ in our case) and $Q(x_i, y_i) = 0$, for all $i \in [n]$.
2. Factor Q into irreducible polynomials (this can be done in polynomial time).

3. Output all polynomials p of degree at most k such that $(y - p(x)) \mid Q(x, y)$ and $p(x_i) = y_i$ for at least a values of $i \in [n]$.

Correctness

The first step is equivalent to a system of n linear equations in the coefficients of Q. There are a variables for the coefficients of monomials of the kind $x^i y^0$, since the $(1, k)$-weighted degree of Q is less than a, i.e. $i + 0k < a$. Similarly there are $a - k$ variables for the coefficients of monomials of the kind $x^i y^1$ and so on. The total number of variables is

$$\begin{aligned} \left\lceil \frac{a}{k} \right\rceil \left(a - \frac{k \left(\lceil \frac{a}{k} \rceil - 1 \right)}{2} \right) &> \frac{a^2}{2k} \\ &= n. \end{aligned}$$

The inequality is true because $\lceil a/k \rceil \geq a/k$ and $\lceil a/k \rceil - 1 < a/k$. Therefore the existence of non-zero solution to the system is guaranteed.

The correctness of the third step, i.e. that it outputs all polynomials of degree at most k that pass through at least a of the given points, is established by the following lemma.

Lemma 2 *If p is a polynomial of degree at most k such that $p(x_i) = y_i$ for at least a values of $i \in [n]$ then $(y - p(x)) \mid Q(x, y)$.*

Proof: Define $f(x) = Q(x, p(x))$. Since $p(x_i) = y_i$ for at least a values of $i \in [n]$, $f(x_i) = Q(x_i, p(x_i)) = Q(x_i, y_i) = 0$ for at least a values of $i \in [n]$. On the other hand since the $(1, k)$-weighted degree of Q is less than a and p is of degree at most k, the degree of f is less than a. Therefore $f \equiv 0$.

$f \equiv 0$ implies that $y = p(x)$ is a root of $Q_x(y)$, where $Q_x(y)$ is $Q(x, y)$ considered as a polynomial from $(\mathbb{F}_q[x])[y]$. Hence $(y - p(x)) \mid Q_x(y) = Q(x, y)$. $\square$