CS761 Derandomization and Pseudorandomness **2022-23 Sem I**

Lecture 12: 22-09-22

Scribe: Punna Hitesh Kumar *Lecturer: Rohit Gurjar*

List Decoding Algorithms

Our goal is to describe efficient list-decoding algorithms for the ReedSolomon codes and its variants. Recall that unique decoding is possible only when the number of errors is assumed to be less than half of the distance of the code. The idea of list decoding is to go beyond half the distance and output all the messages which are within the desired radius. For two words x, y, we will use $\mathrm{agr}(x, y)$ to denote the number of places where x and y agree.

Definition 12.1. *Let C be a code with encoding function Enc: $\Sigma^n \to \Sigma^{n'}$. For any $r \in \Sigma^{n'}$ and a number t, we define List$(r, t) = \{m \in \Sigma^n : agr(Enc(m), r) > t\}$.*

Now, our aim is to find elements belonging to the set $\mathrm{List}(r, t)$, for any word r. We will show that this can be done in polynomial time in the bit length of r.

List-Decoding for ReedSolomon Codes

Problem statement. Fix n evaluation points $\alpha_1, \alpha_2, \dots, \alpha_n \in \mathbb{F}_q$. For any given word $r = (r_1, r_2, \dots, r_n) \in \mathbb{F}_q^n$, we aim to find all polynomials f of degree $< k$ such that $(f(\alpha_1), f(\alpha_2), \dots, f(\alpha_n))$ agrees with r on more than $t = 2\sqrt{nk}$ coordinates. That is, we want to find the set $\mathrm{List}(r, t)$ for the code $\mathrm{RS}(k, n)$, where $t = 2\sqrt{nk}$. As we will see later, the factor "2" in the above bound can be further reduced to 1, which will achieve the Johnson bound (i.e., radius $n - \sqrt{nk}$).

Solution. We aim to find the desired set of polynomials using the Welsh-Berlekamp algorithm.

1. Find a polynomial Q(x,y) $= A_0(x) + A_1(x)y + A_2(x)y^2 + \cdots + A_\ell(x)y^\ell$, such that $Q(\alpha_i, r_i) = 0$ for all $1 \le i \le n$. Here, each i imposes a linear constraint on coefficients.
2. Then we aim to find all factors of $Q(x, y)$ of the form $(y - f(x))$, where degree$(f) < k$ and $f \in \mathit{List}(r, t)$.

Now, the first step is same as solving a system of linear equations, here we will constraint the degree of each $A_i(x) < n/\ell$. Hence, for solving the systems of equations we have total variables $(\ell + 1)(n/\ell)$ which is $> n$ (that is, more than the number of equations). This means that the system has a non-zero solution which can be found in polynomial time by linear algebra over F_q.

For any $f \in \mathbb{F}_q[x]$ of degree $< k$ and agreeing with r at more than $2\sqrt{nk}$, we wish to show that $R(x) := Q(x, f(x)) \equiv 0$. We can say that $\deg(R) \le (k - 1)\ell + n/\ell$. If we show that the number of agreements between f and r is more than degree(R), i.e., $t > (k - 1)\ell + n/\ell$, then we are done. This is because wherever f agrees with r, i.e., $f(\alpha_i) = r_i$, we have $R(\alpha_i) = 0$. The minimum value for RHS is when $\ell = \sqrt{n/(k - 1)}$, and the minimum value is $2\sqrt{n(k - 1)}$.

Finally, $Q(x, f(x)) \equiv 0$ implies that $y - f(x)$ is indeed a factor of $Q(x, y)$. This proves the correctness of the algorithm.

Improving the list decoding radius. The above bound $2\sqrt{n(k - 1)}$ on number of agreements can be further reduced to $\sqrt{2nk}$ in the following way.

1. Let us choose each A_i with degree at most $t_0 - (k - 1)i$, here $t_0 = \sqrt{2n(k - 1)}$, we choose it such way because, in the polynomial $R(x)$, each term $(f(x))^i A_i(x)$ will have the same degree maximum degree t_0. Hence we satisfy the second condition directly by choosing $t = t_0$.

2. For the first condition the total number of coefficients we have is $\Sigma_{i=0}^{i=t_0/(k-1)}(t_0-(k-1)i) \approx t_0^2/(2*(k-1))$, so choosing $t_0 = \sqrt{2n(k-1)}$, will give the number of coefficients as more than n, which will guarantee the existence of a non-trivial solution.

Further improvement. We can further reduce this bound to $t > \sqrt{nk}$ with the help of **method of multiplicities**.

Definition 12.2. *A polynomial $Q(x, y)$ is said to have a zero of multiplicity at least r at (α, β) if for all i, j such that $i + j < r$,*

$$\frac{\partial Q}{\partial x^i \partial y^j}(\alpha, \beta) = 0$$

Application.

1. Choose numbers D, s such that $D/s = \sqrt{n(k-1)}$. Find a nonzero polynomial $Q(x, y) = A_0(x) + A_1(x)y + A_2(x)y^2 + \cdots + A_\ell(x)y^\ell$ with $\ell = D/(k-1)$ with each degree$(A_i) < D - (k-1)i$ such that each (α_i, r_i) is a root of $Q(x, y)$ with multiplicity s.

2. We find all polynomials $f(x)$ such that $y - f(x)$ is a factor of $Q(x, y)$, $\deg(f) < k$ and $f \in List(r, t)$, where t will be set as D/s.

Analysis. First let us argue that a nonzero solution $Q(x, y)$ satisfying the desired constraints can be found. The number of unknowns is again $\approx D^2/(2(k-1))$ like the previous approach. But, the number of equations gets amplified i.e., we have roughly $\binom{s+1}{2}n$. This is because for each (α_h, r_h) and for each i, j with $i + j < s$, we will put the constraint

$$\frac{\partial Q}{\partial x^i \partial y^j}(\alpha_h, r_h) = 0.$$

Now, to ensure that the number of unknowns is more than the number of equations, we need

$$D^2/(2(k-1)) \geq \binom{s+1}{2}n$$

equivalently, $D^2/s^2 \geq n(k-1)$.

Now, we argue that for any $f(x)$ with degree $< k$ which agrees with r on more than t coordinates, $y - f(x)$ is indeed a factor of $Q(x, y)$. Due to our additional multiplicity constraints, if $f(\alpha_i) = r_i$, then $R(x) = Q(x, f(x))$ vanishes with multiplicity at least s times at $alpha_i$. That means, $R(x)$ has more than t roots, each with multiplicity s. But, if ts is more than the degree of $R(x)$, which is $< D$, then $R(x) = Q(x, f(x))$ is zero as a polynomial. Hence, $y - f(x)$ is a factor of $Q(x, y)$. Thus, our choice of $t = D/s = \sqrt{n(k-1)}$ works well.