

Lecture 37: List Decoding of RS

November 27, 2007

Lecturer: Atri Rudra

Scribe: Thanh-Nhan Nguyen & Atri Rudra

1 List Decoding Algorithm

Recall that list decoding capacity is $1 - H_q(p)$. And we know that there exists an $(p, O(\frac{1}{\varepsilon}))$ -list decodable code of rate $1 - H_q(p) - \varepsilon$. Following are some natural questions

1. Are there explicit codes that achieve list decoding capacity $1 - H_q(p)$ with efficient list decoding algorithms?

Recall that $1 - H_q(p) = 1 - p - \varepsilon$ for $q = 2^{\Omega(\frac{1}{\varepsilon})}$

2. Are there explicit codes with rate $R > 0$ that can list-decode up to $1 - R - \varepsilon$ fraction of errors with efficient list decoding algorithms?

Johnson bound with alphabet free version:

For any code with distance δ , it is a $(J_q(\delta), O(n^2))$ list decodable

$$J_q(\delta) = 1 - \sqrt{1 - \delta}$$

Then, explicitness is not an issue.

By Singleton bound

$$O(1) + 1 - \delta \geq R$$

Then

$$J_q(\delta) \leq 1 - \sqrt{1 - R}$$

3. Is there an efficient list decoding algorithm for code of rate $R > 0$ that can correct $1 - \sqrt{1 - R}$ fraction of errors?

2 Question 3

Consider any $[n, k + 1]_q$ RS codes.

Input: Received word (α_i, y_i) , $\alpha_i, y_i \in F_q$, error parameter $e = n - t$

Output: All polynomials $P(X) \in F_q[X]$ of degree at most R such that $p(\alpha_i) = y_i$ for at least t values of i .

Goal: Make t as small as possible.

2.1 Berlekamp Welch Algorithm

$$(t > \frac{n+k}{2})$$

- Step 1: Find polynomials $B(X)$ of degree $k + e$, and $N(X)$ of degree e such that

$$\forall 1 \leq i \leq n, B(\alpha_i) = y_i N(\alpha_i)$$

- Step 2: Show that $Y - P(X)$ divides $Q(X, Y)$

2.2 Structure of list decoding algorithms for RS

- Step 1: (Interpolation) Find non-zero $Q(X, Y)$ such that $Q(\alpha_i, y_i) = 0, 1 \leq i \leq n$
- Step 2: If $P(X)$ needs to be output then $Y - P(X)$ is a factor of $Q(X, Y)$ (root finding)

There is a fact that bivariate polynomials can be factored efficiently.

- Step 1: Solve for coefficients of $Q(X, Y)$. This can be done as long as the number of coefficients are greater than 0
- Step 2: $Q(X, Y)$ needs to be restricted

We recall the definition of maximum degree of a variable.

Definition 2.1. $\deg_X(Q)$ is the maximum degree of X in $Q(X, Y)$. Similarly, $\deg_Y(Q)$ is the maximum degree of Y in $Q(X, Y)$

Given $\deg_X(Q) = a$ and $\deg_Y(Q) = b$, we have

$$Q(X, Y) = \sum_{0 \leq i \leq a} q_{ij} X^i Y^j$$

In above formula, the number of coefficients is equal to $(a + 1)(b + 1)$. We want this product greater than n .

2.3 Algorithm 1

- Step 1: Find a non-zero $Q(X, Y)$ with $\deg_X(Q) \leq l, \deg_Y(Q) \leq \frac{n}{l}$ for some l to be fixed later such that

$$Q(\alpha_i, y_i) = 0, 1 \leq i \leq n$$

- Step 2: Output $P(X)$ if $Y - P(X)$ divides $Q(X, Y)$

Validity of step 1:

$$\#coeffs = (l + 1)\left(\frac{n}{l} + 1\right) > n$$

Validity of step 2: we need to show that if $P(X)$ of degree $\leq k$ agrees with Y in at least t position then $Y - P(X)$ divides $Q(X, Y)$

$$R(X) \triangleq Q(X, P(X))$$

We need to show $R(X) \equiv 0$

$$\deg(R) \leq \deg_X(Q) + \deg P \deg_Y(Q) \tag{1}$$

$$\leq l + \frac{nk}{l} \tag{2}$$

If $P(\alpha_i) = y_i$ then

$$Q(\alpha_i, y_i) = Q(\alpha_i, P(\alpha_i)) = 0$$

Thus α_i is a root of $R(X)$. So, R has at least t roots.

The proof is done if $t > l + \frac{nk}{l}$.

We pick $l = \sqrt{nk}$, this implies $t > 2\sqrt{nk}$. Then, Algorithm 1 works. Thus the fraction of errors that can be handled is $1 - 2\sqrt{\frac{k}{n}} = 1 - 2\sqrt{R}$.