Discussion 5

March 8, 2019

1 Polynomials

1.1 Property

A non-zero polynomial of degree d has at most d roots.

1.2 Interpolation

Given $d+1$ pairs $(x_1, y_1), \ldots, (x_{d+1}, y_{d+1})$ with all the x_i distinct, there exists a unique polynomial $p(x)$ of degree (at most) d such that $p(x_i) = y_i$ for $a \leq i \leq d+1$. $p(x)$ can be found by Lagrangian method.

2 Error Correcting Codes

2.1 Erasure Errors

To send a message of length n and with at most k loss of packages, we need to send $n+k$ packages.

2.2 General Errors

The setting of the problem is that we want to send a message of length n through packages which contain one character in each, and there will be at most k corruption on packages before trasmission completes. The claim is that if we send out $n+2k$ packages, we can for sure recover the message. I want to split the discussion on general erros into two parts: i) the computation process of message recovering, and ii) why this mechanics gives the correct message.

2.2.1 Computation

- We set up a polynomial $Q(x)$ of degree $n+k-1$, and a polynomial $E(x)$ of degree k. The degree of these two polynomials are all that we know for now, so we write $Q(x) = a_{n+k-1}x^{n+k-1} + a_{n+k-2}x^{n+k-1} + \cdots + a_1 x + a_0, E(x) = x^k + b_{k-1}x^{k-1} + \cdots + b_1 x + b_0$. Notice that I have $n+k$ unknown coefficients for $Q(x)$ and k for $E(x)$ because I implicitly scaled $E(x)$ so that the coefficient for x^k is one.

- Another piece we know is the characters in the packages received. There $n + 2k$ of them, denoted as $r_1, r_2, \ldots, r_{n+2k}$ which are elements in modular field $\mathbb{Z}/\mathbb{Z}_q$. Then I write out $n + 2k$ equations:

$$Q(1) = r_1 E(1)$$

$$Q(2) = r_2 E(2)$$

$$\ldots$$

$$Q(n + 2k) = r_{n+2k} E(n + 2k)$$

Or more explicitly:

$$a_{n+k-1} + a_{n+k-2} + \cdots + a_0 = r_1(1 + b_{k-1} + b_{k-2} + \cdots + b_0)$$

$$a_{n+k-1}2^{n+k-1} + a_{n+k-2}2^{n+k-2} + \cdots + a_0 = r_1(2^k + b_{k-1}2^{k-1} + b_{k-2}2^{k-2} + \cdots + b_0)$$

$$\ldots$$

$$a_{n+k-1}(n+2k)^{n+k-1} + a_{n+k-2}(n+2k)^{n+k-2} + \cdots + a_0 = r_1((n+2k)^k + b_{k-1}(n+2k)^{k-1} + b_{k-2}2^{k-2} + \cdots + b_0)$$

- Now solve the above $n + 2k$ equations with $n + 2k$ variables. Don't forget that they are computed in $\mod q$.

- Compute $\frac{Q(x)}{E(x)}$. I claim that the result polynomial is exactly $P(x)$.

2.2.2 Proof

- Firstly I define $E'(x) = (x - x_1)(x - x_2)\ldots(x - x_k), Q'(x) = P(x)E(x), x = 1, 2, \ldots, n + 2k$ where $x_1, x_2, \ldots, x_k$ are index of corrupted packages and $P(x)$ is the polynomial that encrypts the message.

- As verified in the discussion worksheet, $E'(x), Q'(x)$ are valid solutions to the linear equations.

- Since $\frac{Q(x)}{E(x)}$ and $\frac{Q'(x)}{E'(x)} = P(x)$ gives the same polynomial, we know that the polynomial we recovered is always correct.