CS70–Spring 2013 — Homework 6

Felix He, SID 303308****

April 11, 2024

Collaborators: None

1. d+2 points vs. a polynomial of degree d

1.

$$\Delta_0 x = \frac{(x-1)(x-2)}{(0-1)(0-2)} = \frac{(x-1)(x-2)}{2}$$
$$\Delta_1 x = \frac{(x-0)(x-2)}{(1-0)(1-2)} = -x(x-2)$$
$$\Delta_2 x = \frac{(x-0)(x-1)}{(2-0)(2-1)} = \frac{x(x-1)}{2}$$

$$p(x) = 1 \cdot \Delta_0 x + 1 \cdot \Delta_1 x + 3 \cdot \Delta_2 x = x^2 - x + 1$$

2. Yes. $p(-1) = 3$

3. No. $p(-1) = 3 \neq 0$.
As we know from property 2, $d+1$ pairs of (x, y) uniquely determine a polynomial $p(x)$ of degree at most d. So we know $p(x)$ from part 1 passes through 3 points (0, 1), (1, 1), and (2, 3), then we know it's the unique polynomial, if $p(x_4) \neq y_4$, we know the fourth point is not on the polynomial. So any of 3 of 4 points construct a unique polynomial $p'(x)$, if all points are on it, we know it must be the same with $p(x)$, however, we know $p(x)$ does not pass the fourth point, so it's a contradiction.

4. Pseudocode:

```
def fit_polynomials(x1, y1, x2, y2, x3, y3, x4, y4):
    if not check_disctinct(x1, x2, x3, x4):
        raise Error("x_i not distinct")
    else:
        poly = LagrangeInter(x1, y1, x2, y2, x3, y3)
        if eval_poly(poly, x4) == y4:
            return True
        else:
            return False
```

2. Working with polynomials

1. $3^3 = 27$ possible functions. e.g. One possible function is $f(0) = 0, f(1) = 0, f(2) = 0$

2. 3 possible functions.

$$f(x) = 0$$
$$f(x) = 1$$
$$f(x) = 2$$

3. $2 * 3 = 6$ possible functions. $f(x) = ax + b$. a can take 2 values (not 0), b can take 3 values.

$$f(x) = x$$
$$f(x) = x + 1$$
$$f(x) = x + 2$$
$$f(x) = 2x$$
$$f(x) = 2x + 1$$
$$f(x) = 2x + 2$$

4. $2 * 3^3 = 18$ possible functions. $f(x) = ax^2 + bx + c$. (a can take 1 and 2)

$$f(x) = x^2$$
$$f(x) = x^2 + x$$
$$f(x) = x^2 + 2x$$
$$f(x) = x^2 + 1$$
$$f(x) = x^2 + x + 1$$
$$f(x) = x^2 + 2x + 1$$
$$f(x) = x^2 + 2$$
$$f(x) = x^2 + x + 2$$
$$f(x) = x^2 + 2x + 2$$
$$f(x) = 2x^2$$
$$f(x) = 2x^2 + x$$
$$f(x) = 2x^2 + 2x$$
$$f(x) = 2x^2 + 1$$
$$f(x) = 2x^2 + x + 1$$
$$f(x) = 2x^2 + 2x + 1$$
$$f(x) = 2x^2 + 2$$
$$f(x) = 2x^2 + x + 2$$
$$f(x) = 2x^2 + 2x + 2$$

5. It's an one-to-one mapping from degree 0,1,2 polynomials to functions in part 1. 3 pts determine a unique polynomial. So GF(3)-GF(3) has 27 possible combinations of $((x1, y1), (x2, y2), (x3, y3))$, so each combination uniquely determine one polynomial.

Polynomial	**Mapping**
$f(x) = 0$	$(0, 0, 0)$
$f(x) = 1$	$(1, 1, 1)$
$f(x) = 2$	$(2, 2, 2)$
$f(x) = x$	$(0, 1, 2)$
$f(x) = x + 1$	$(1, 2, 0)$
$f(x) = x + 2$	$(2, 0, 1)$
$f(x) = 2x$	$(0, 2, 1)$
$f(x) = 2x + 1$	$(1, 0, 2)$
$f(x) = 2x + 2$	$(2, 1, 0)$
$f(x) = x^2$	$(0, 1, 1)$
$f(x) = x^2 + 1$	$(1, 2, 2)$
$f(x) = x^2 + 2$	$(2, 0, 0)$
$f(x) = x^2 + x$	$(0, 2, 0)$
$f(x) = x^2 + x + 1$	$(1, 0, 1)$
$f(x) = x^2 + x + 2$	$(2, 1, 2)$
$f(x) = x^2 + 2x$	$(0, 0, 2)$
$f(x) = x^2 + 2x + 1$	$(1, 1, 0)$
$f(x) = x^2 + 2x + 2$	$(2, 2, 1)$
$f(x) = 2x^2$	$(0, 2, 2)$
$f(x) = 2x^2 + 1$	$(1, 0, 0)$
$f(x) = 2x^2 + 2$	$(2, 1, 1)$
$f(x) = 2x^2 + x$	$(0, 0, 1)$
$f(x) = 2x^2 + x + 1$	$(1, 1, 2)$
$f(x) = 2x^2 + x + 2$	$(2, 2, 0)$
$f(x) = 2x^2 + 2x$	$(0, 1, 0)$
$f(x) = 2x^2 + 2x + 1$	$(1, 2, 1)$
$f(x) = 2x^2 + 2x + 2$	$(2, 0, 2)$

Table 1: Polynomials Functions Mapping

6. it's the same function with $f(x) = x$

7. First take any coefficients mod 3 first.
   Second any degree > 2 terms, we reduce it to degree ≤ 2. $x^{3-1} \mod 3 = 1 \mod 3, x \in 1, 2$, according to Fermat's Little Theorem. And since the reduce term is either x or x^2, it remains the same value when $x = 0$
   For example,
   Consider a polynomial in $GF(3)$:

   $$f(x) = 2x^4 + x^3 + 2x + 1$$

   Reduce Powers: Since $x^4 \equiv x^2$ and $x^3 \equiv x$ in $GF(3)$, we have:

   $$f(x) \equiv 2x^2 + x + 2x + 1$$

   Combine Like Terms: Coefficients can be added modulo 3, so $3x \equiv 0x$, giving us:

   $$f(x) \equiv 2x^2 + 1$$

The transformed polynomial is thus:

$$g(x) = 2x^2 + 1$$

3. How many errors?

$n + 2k \leq 25$, k at most 7. So at most 7 errors can we recover from.

4. Linearity of Reed-Solomon codes

1. For any i (i is the position of the message), we have $m_i = 0$, so $P(i) = m_i = 0$.
2. Use Lagrange Interpolation, we know, $\Delta_i(x)$ takes 1 at $x = i$, and 0 at $x \neq i$, So if $P(i) = m$, it means $P(i) = m \cdot \Delta_i(x)$. if $P'(i) = m' = \alpha m$, it means $P'(i) = \alpha \cdot m \cdot \Delta_i(x) = \alpha \cdot P(i)$
3. Use Lagrange Interpolation.

$$\begin{aligned} P(i) = m, \;\; P(i) = m \cdot \Delta_i(x) \\ P'(i) = m', \;\; P'(i) = m' \cdot \Delta_i(x) \\ P''(i) = m'', \;\; P''(i) = m'' \cdot \Delta_i(x) \end{aligned}$$

And we know

$$m'' = m + m'$$

So

$$P''(i) = m'' \cdot \Delta_i(x) = (m + m')\Delta_i(x) = m \cdot \Delta_i(x) + m' \cdot \Delta_i(x) = P(i) + P'(i)$$

5. Again, find the liar

1. No. degree-2 polynomial is determined by 3 points. So if one person lies, any 3 points out of 4 can construct a polynomial and the extra pt would not be on it. So you won't know which one lie.
2. Yes. $n + 2k = 5, n = 3, k = 1$. You know one person lies, the other four don't. So you can find a polynomial using 3 points and see is there another point on it. If there are four points on it, then it's the correct polynomial, and you can also figure out which point is not on it, then it's the liar.
3. Yes. If the machine output YES, then no liar. If output NO then there is one liar. Use the General Error Detecting Codes. In this case,

$$\begin{aligned} E(x) &= (x - e_1) \\ P(i)E(i) &= r_i E(i) \quad \text{for } 1 \leq i \leq n + 2k = d + 1 + 2 = d + 3 \end{aligned}$$

Here we know, $k = 1$, $n = d + 1$, $deg(E(x)) = 1$, $deg(P(x)) = d$

$$Q(x) := P(x)E(x)$$

Then $deg(Q(x)) = d + 1$ So we know there are $d + 3$ people for at most degree d polynomial and $k = 1$ error, we can detect the liar with the machine

4. We use $Q(x) = P(x)E(x)$, $P(i)E(i) = r_i E(i) \quad \text{for} 1 \leq i \leq n + 2k$
   We can have:

$$E(x) = (x - e) = (x + d)$$
$$Q(x) = ax^2 + bx + c$$

   Then we can have $a = -1, b = 7, c = -12, d = -3$

$$Q(x) = (-x^2 + 7x - 12)$$
$$E(x) = (x - 3)$$
$$P(x) = Q(x)/E(x) = -x + 4$$

5. Just 1 time. You can compute the $E(x)$ just in one time to figure out who is the liar.

6. Construct a message

Use Lagrange interpolation to find the polynomial $p(x)$ by $(0, 4), (1, 3), (2, 2)$

$$\begin{aligned} p(x) &= 4 \cdot \frac{(x-1)(x-2)}{(0-1)(0-2)} + 3 \cdot \frac{(x-0)(x-2)}{(1-0)(1-2)} + 2 \cdot \frac{(x-0)(x-1)}{(2-0)(2-1)} \\ &= -x + 4 \end{aligned}$$

$P(3) = 1, P(4) = 0$

7. Trust No One

This involves a hierarchy of sharing.

Since we need two races to agree unlock the secret. We first pick a degree-1 polynomial $p(x) = ax + s$(a is picked by us), to generate the $p(1), p(2), p(3), p(4)$ for the 4 races. $p(0) = s$ is the secret. And then $p(1), p(2), p(3), p(4)$ become secrets for each races, meaning they need unanimous agreement to unlock it. Then we do the similar things again.

For example, the hobbits have 4 members, we randomly pick 3 points $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ and combine with $(0, p(1))$, 4 points determine a degree-3 polynomial $f_1(x)$, then we use it to calculate for the last member $y_4 = f_1(x_4)$, then we send these four points $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$ to 4 hobbits. So only if 4 of them agree, they can construct the polynomial $f_1(x)$ and calculate $p(1) = f_1(0)$. The same with other races. As long as we get to $p(i)$, we can reconstruct the original polynomial $p(x)$, therefore to get $s = p(0)$.

8. Multiplying Polynomials Quickly? Almost...

1.

$$c_i = \sum_{k=0}^{i} a_k b_{i-k} \quad \text{for } 0 \leq x \leq n \quad c_i = \sum_{k=n+1}^{2n} a_k b_{i-k} \text{for } n < x \leq 2n$$

2. multiplications

   (a) $i+1$ if $i \leq n$

   (b) $n-(i-n)+1 = 2n-i+1$ if $i > n$

   additions, since adding 2 terms need 1 additions, so

   (a) i if $i \leq n$

   (b) $n-(i-n)+1 = 2n-i$ if $i > n$

3. multiplications

$$\sum_{i=0}^{n} i+1+\sum_{i=n+1}^{2n}(2n-i+1) = n^2+2n+1$$

   additions:

$$\sum_{i=0}^{n} i+\sum_{i=n+1}^{2n}(2n-i) = n^2$$

4. Multiplications $\frac{n(n+1)}{2}$, ($i-1+1$ multiplications for term i, and n terms need multiplications). Additions n
   $i-1+1$ multiplications for term i, and n terms need multiplications. Adding all $n+1$ terms take n time.

5. (a) Multiplications: $\frac{n(n+1)}{2} * (2n+1) * 2 = \Theta(n^3)$ for P and Q together. Additions: $\frac{n(n+1)}{2} * n * 2 = \Theta(n^3)$

   (b) calculate $P(x)Q(x)$ for each x takes $2n+1$ time

   (c) $\Theta(n^2)$. Since each Δ_i takes $2n$ multiplications, and there are $2n+1$ points

6. Slower, one is $\Theta(n^3)$, part 3 is $\Theta(n^2)$

7. n multiplications and n additions

8. Similar, we just substitute part7 for part4, reduce the n^2 to n, finally the time is $\Theta(n^2)$

9. Similar, they are both $\Theta(n^2)$