Polynomials, Secret Sharing, and Error-Correcting Codes

Lecture 9 (February 14, 2006)

Remember the first assignment of the class?

Each person obtained a character and a position

You had to come together as a class to determine a shared secret

It took only 1/3 of the class to determine the secret

Today we will see how to split a secret into n “shares” and guarantee that the secret can only be determined if k out of n people come together

Polynomials

$P(x) = 3x^2 + 7x – 2$

$Q(x) = x^{123} – 0.5x^{25} + 19x^3 – 1$

$S(z) = z^2 – z - 1$

$W(x) = \pi$

Representing A Polynomial

A degree-d polynomial is represented by its (d+1) coefficients

$$P(x) = a_d x^d + a_{d-1} x^{d-1} + \ldots + a_1 x^1 + a_0$$

The numbers $a_d, a_{d-1}, \ldots, a_0$ are the coefficients

E.g. $P(x) = 3x^4 – 7x^2 + 12x – 19$

Coefficients are: 3, 0, -7, 12, -19

What Are The Coefficients?

The coefficients could be real numbers, integers, rational numbers, etc

In this lecture, we will work with coefficients from Z_p (where p is a prime number)

$$Z_p = \{0, 1, 2, \ldots, p-1\}$$

Facts About Polynomials

Let P(x), Q(x) be two polynomials

The sum P(x)+Q(x) is also a polynomial
(i.e., polynomials are "closed under addition")

Their product P(x)Q(x) is also a polynomial
("closed under multiplication")

P(x)/Q(x) is not necessarily a polynomial

Multiplying Polynomials

$$(x^2+2x-1)(3x^3+7x) = 3x^5 + 7x^3 + 6x^4 + 14x^2 - 3x^3 - 7x$$
$$= 3x^5 + 6x^4 + 4x^3 + 14x^2 - 7x$$

Evaluating A Polynomial

Suppose: $P(x) = 3x^4 - 7x^2 + 12x - 19$

$P(0) = -19$

$P(5) = 3\times5^4 - 7\times5^2 + 12\times5 - 19$

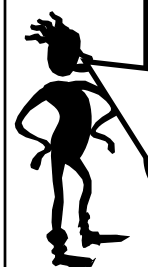
$P(-1) = 3\times(-1)^4 - 7\times(-1)^2 + 12\times(-1) - 19$

The Roots of a Polynomial

Suppose: $P(x) = a_d x^d + a_{d-1} x^{d-1} + \ldots + a_1 x^1 + a_0$

Definition: r is a “root” of P(x) if P(r) = 0

$P(x) = 3x + 7$	root = - (7/3)
$P(x) = x^2 - 2x + 1$	root = 1
$P(x) = 3x^3 - 10x^2 + 9x - 2$	roots = 1/3, 1, 2



The Single Most Important Fact About Low-degree Polynomials

A non-zero degree-d polynomial P(x) has at most d roots

A Crucial Implication

Assume P(x) and Q(x) have degree at most d

Suppose $x_1, x_2, \ldots, x_{d+1}$ are d+1 points such that $P(x_k) = Q(x_k)$ for all k = 1,2,…,d+1

Then P(x) = Q(x) for all values of x

Proof: Define R(x) = P(x) – Q(x)

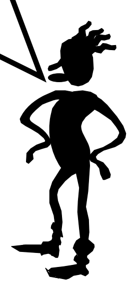
R(x) has degree d

R(x) has d+1 roots, so it must be the zero polynomial

If you give me pairs $(x_1, y_1), \ldots, (x_{d+1}, y_{d+1})$

then there is at most one degree-d polynomial P(x) such that:

$P(x_k) = y_k$ for all k



Hmm: at most one?

So perhaps there are no such degree-d polynomials with

$P(x_k) = y_k$

for all the d+1 values of k

Lagrange Interpolation

Given (d+1) pairs $(x_1, y_1), (x_2, y_2), \ldots, (x_{d+1}, y_{d+1})$

then there is EXACTLY ONE degree-d polynomial P(x) such that

$P(x_k) = y_k$ for all k

k-th "Switch" polynomial

Given (d+1) pairs $(x_1, y_1), (x_2, y_2), \ldots, (x_{d+1}, y_{d+1})$

$g_k(x) = (x-x_1)(x-x_2)\ldots(x-x_{k-1})(x-x_{k+1})\ldots(x-x_{d+1})$

Degree of $g_k(x)$ is: d

$g_k(x)$ has d roots: $x_1,\ldots,x_{k-1},x_{k+1},\ldots,x_{d+1}$

$$h_k(x) = \frac{(x-x_1)(x-x_2)\ldots(x-x_{k-1})(x-x_{k+1})\ldots(x-x_{d+1})}{(x_k-x_1)(x_k-x_2)\ldots(x_k-x_{k-1})(x_k-x_{k+1})\ldots(x_k-x_{d+1})}$$

$h_k(x_k) = 1$

For all $i \neq k$, $h_k(x_i) = 0$

The Lagrange Polynomial

Given (d+1) pairs $(x_1, y_1), (x_2, y_2), \ldots, (x_{d+1}, y_{d+1})$

$$P(x) = y_1h_1(x) + y_2h_2(x) + \ldots + y_{d+1}h_{d+1}(x)$$

$$h_k(x) = \frac{(x-x_1)(x-x_2)\ldots(x-x_{k-1})(x-x_{k+1})\ldots(x-x_{d+1})}{(x_k-x_1)(x_k-x_2)\ldots(x_k-x_{k-1})(x_k-x_{k+1})\ldots(x_k-x_{d+1})}$$

P(x) is the unique polynomial of degree d such that $P(x_1) = y_1, P(x_2) = y_2, \ldots, P(x_{d+1}) = y_{d+1}$

Example

Input: (0,1), (1,2), (2,9)

Switch polynomials:

$h_1(x) = (x-1)(x-2)/(0-1)(0-2) = ½ (x-1)(x-2)$

$h_2(x) = (x-0)(x-2)/(1-0)(1-2) = x(x-2)/(-1)$

$h_3(x) = (x-0)(x-1)/(2-0)(2-1) = ½ x(x-1)$

$P(x) = 1 \times h_1(x) + 2 \times h_2(x) + 9 \times h_3(x)$

$= 3x^2 – 2x + 1$

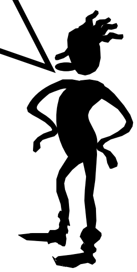
To recap:

If you give me pairs
$(x_1, y_1), (x_2, y_2), \ldots, (x_{d+1}, y_{d+1})$

then there is exactly one
degree-d P(x) such that

$P(x_k) = y_k$ for all k

(And I can find this P(x)
using Lagrange interpolation)



Two Different Representations

$P(x) = a_d x^d + a_{d-1}x^{d-1} + \ldots + a_1 x^1 + a_0$
can be represented either by

1. d+1 coefficients $a_d, a_{d-1}, \ldots, a_2, a_1, a_0$
2. Its value at any d+1 points $P(x_1), \ldots, P(x_d), P(x_{d+1})$ (e.g., P(0), P(1), P(2), …, P(d+1))

Converting Between The Two Representations

Coefficients to Evaluation:

Evaluate P(x) at d+1 points

Evaluation to Coefficients:

Use Lagrange Interpolation

Difference In The Representations

P(x) can be represented by:
- a) d+1 coefficients $a_d, a_{d-1}, \ldots, a_1, a_0$
- b) Value at d+1 points $P(x_1), \ldots, P(x_{d+1})$

Adding two polynomials:

Both representations are equally good, since in both cases the new polynomial can be represented by the sum of the representations

Difference In The Representations

P(x) can be represented by:
- a) d+1 coefficients $a_d, a_{d-1}, \ldots, a_1, a_0$
- b) Value at d+1 points $P(x_1), \ldots, P(x_{d+1})$

Multiplying two polynomials:

Representation (a) requires $(d+1)^2$ multiplications

Representation (b) requires finding new points in both polynomials

Difference In The Representations

P(x) can be represented by:
- a) d+1 coefficients $a_d, a_{d-1}, \ldots, a_1, a_0$
- b) Value at d+1 points $P(x_1), \ldots, P(x_{d+1})$

Evaluating the polynomial at some point:

Is easy with representation (a)

Requires Lagrange interpolation with (b)

The Value-Representation Is Tolerant To "Erasures"

I want to send you a polynomial P(x) of degree d

Suppose your mailer drops my emails once in a while

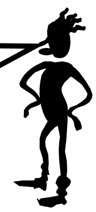
The Value-Representation Is Tolerant To "Erasures"

I want to send you a polynomial $P(x)$ of degree d

Suppose your mailer drops my emails once in a while

Say, I wanted to say "hello". I could write it as "8 5 12 12 15"

and hence as
$8x^4 + 5x^3 + 12x^2 + 12x + 15$



The Value-Representation Is Tolerant To "Erasures"

I want to send you a polynomial $P(x)$ of degree d

Suppose your mailer drops my emails once in a while

I could evaluate $P(x)$ at $n > d+1$ points and send $(k, P(k))$ to you for all $k = 1, \ldots, d, \ldots, n$

As long you get at least $(d+1)$ of these, you can reconstruct $P(x)$

But Is It Tolerant To "Corruption"?

I want to send you a polynomial $P(x)$ of degree d

Suppose your mailer corrupts my emails once in a while

E.g., suppose $P(x) = 2x^2 + 1$, and I chose $n = 4$

So I sent you (0,1), (1, 3), (2, 9), (3,19)

Corrupted email says (0,1), (1, 2), (2, 9), (3, 19)

You choose (0,1), (1,2), (2,9)
and get $Q(x) =$

Error-Detecting Representation

The above scheme does detect errors!

If we send the value of degree-d polynomial $P(x)$ at $n \geq d+1$ different points,

$$(x_1, P(x_1)), (x_2, P(x_2)), \ldots, (x_n, P(x_n))$$

then we can detect corruptions as long as there are fewer than $(n-d)$ of them

Why? If only n-d-1 corruptions, then d+1 correct points!

Also Error Correcting Representation

As long as fewer than (n-d)/2 corruptions then we can get back the original polynomial P(x)!!!

Error Correcting Codes (ECCs)

Don't need to know which ones are corrupted, just that there are < (n-d)/2 corruptions

(We won't go over how to do this)

Secret Sharing

Missile has random secret number S encoded into its hardware. It will not arm without being given S

n officers have memorized a private, individual "share"

Any k out of n of them should be able to assemble their shares so as to obtain S

Any $\leq$ k-1 of them should not be able to jointly determine any information about S

A k-out-of-n Secret Sharing Scheme

Let S be a random "secret" from Z_p

Want to give shares $S_1, S_2, \ldots, S_n$ to the n officers such that:

- If we have k of the S_i's, then we can find out S
- If we have k-1 S_i's, then any secret is equally likely to have produced this set of S_i's

Our k-out-of-n S.S.S.

Let S be a random "secret" from Z_p

Pick k-1 random coefficients $R_1, R_2, \ldots, R_{k-1}$ from Z_p

Let $P(x) = R_{k-1} x^{k-1} + R_{k-2} x^{k-2} + \ldots + R_1 x^1 + S$

For any j in {1,2,…,n}, officer j's share $S_j = P(j)$

P(0) = where P hits y-axis = S

P(x) chosen to be a random degree k-1 polynomial given that P hits the y-axis at S

Our k-out-of-n S.S.S.

Let S be a random "secret" from Z_p

Pick k-1 random coefficients $R_1, R_2, \ldots, R_{k-1}$ from Z_p

Let $P(x) = R_{k-1} x^{k-1} + R_{k-2} x^{k-2} + \ldots + R_1 x^1 + S$

For any j in {1,2,…,n}, officer j's share $S_j = P(j)$

If k officers get together, they can figure out P(x)
And then evaluate P(0) = S

Our k-out-of-n S.S.S.

If k-1 officers get together, they know P(x) at k-1 different points

For each value of S', we can get a unique polynomial P' passing through their points, and P'(0) = S'

And so each S' equally likely!

Study Bee

Polynomials

Degree-d polynomial has at most d roots

Lagrange Interpolation

Given d+1 pairs (x_k, y_k), can find unique polynomial P such that $P(x_k) = y_k$ for all these k

Applications

Error detecting/correcting codes
Secret sharing