Great Theoretical Ideas In Computer Science

Anupam Gupta

CS 15-251 Fall 2006

Lecture 16 Oct 19, 2006

Carnegie Mellon University

Polynomials, Secret Sharing, And Error-Correcting Codes

P(X) =  X^3 +  X^2 +  X^1 + 

Polynomials in one variable over the reals

$P(x) = 3x^2 + 7x - 2$

$Q(x) = x^{123} - \frac{1}{2} x^{25} + 19 x^3 - 1$

$R(y) = 2y + \sqrt{2}$

$S(z) = z^2 - z - 1$

$T(x) = 0$

$W(x) = \pi$

Representing a polynomial

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

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

The numbers $a_d, a_{d-1}, \dots, a_0$ are the <u>coefficients</u>.

E.g. $P(x) = 3x^4 - 7x^2 + 12x - 19$

Coefficients are: 3, 0, −7, 12, −19

Are we working over the reals?

We could work over any "field"
(set with addition, multiplication, division defined.)

E.g., we could work with the rationals, or the reals.

Or with Z_p, the integers mod prime p.

In this lecture, we will work with Z_p

The Set Z_p for prime p

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

$Z_p^* = \{1, 2, 3, \ldots, p-1\}$

Simple Facts about Polynomials

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

$$\begin{array}{r} 2x^2 + 3x + 5 \\ x^2 - 6x + 9 \\ \hline 3x^2 - 3x + 14 \end{array}$$

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:

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

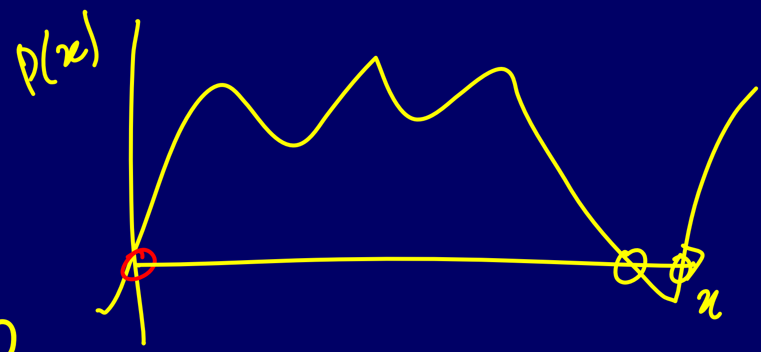
E.g. $P(x) = 3x^4 - 7x^2 + 12x - 19$

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

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

$P(0) = -19$

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

E.g., $P(x) = 3x + 7$ root = $-(7/3)$.

$P(x) = x^2 - 2x + 1$ roots = 1, 1

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

Linear Polynomials

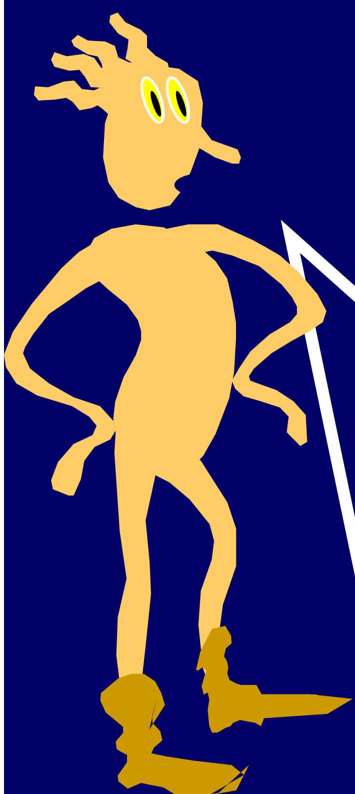
$P(x) = ax + b$

E.g., $P(x) = 7x - 9$ (over Z_{11})

One root: $P(x) = ax + b = 0 \Rightarrow x = -b/a$

E.g., root = (- (-9)/7) = 9 * 7^{-1}
= 9 * 8 = 72
= 6 (mod 11).

Check: P(6) = 7*6 – 9 = 42 – 9 = 33 = 0 (mod 11)



The Single Most Important Fact About Low-degree Polynomials

A <u>non-zero</u> degree-d polynomial P(x) has at most d roots.

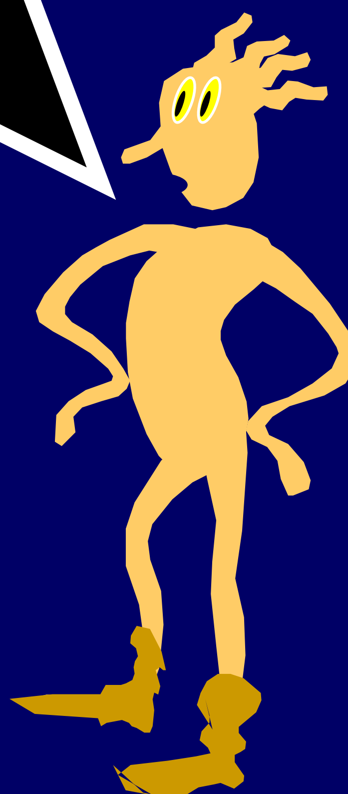
Very Important

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



Why?

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

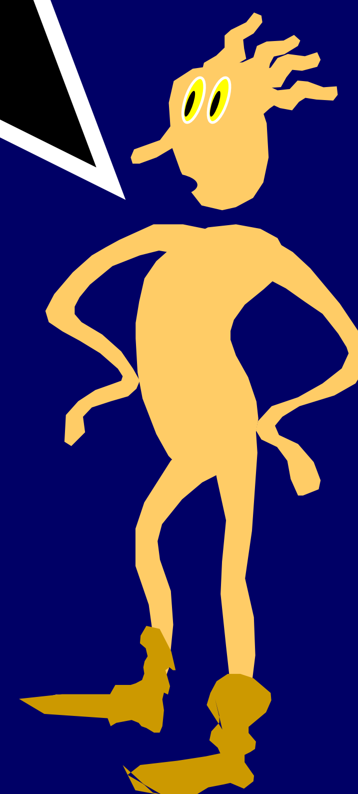
R(x) has d+1 roots, so it must be the zero polynomial

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

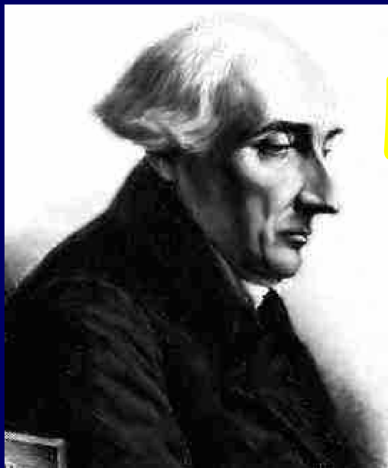


Hmm: at most one?

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

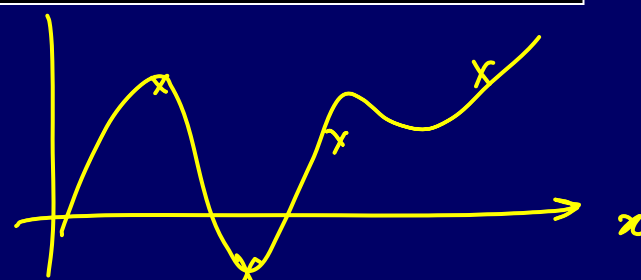
$x_i \neq x_j \quad \forall\, i \neq j$

Given any (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

degree at most d



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: $x_1,\ldots,x_{k-1},x_{k+1},\ldots,x_{d+1}$

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

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

k-th "Switch" polynomial

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

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

$g_k(x)$

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

$h_k(x_k) = 1$

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

$h(x_1) = h(x_2) \ldots$

$= h(x_{k-1})$

$= h(x_{k+1})$

$\ldots = h(x_{d+1}) = 0$

The Lagrange Polynomial

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

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

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

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: (5,1), (6,2), (7,9)

$x_1 = 5$ $x_2 = 6$ $x_3 = 7$
$y_1 = 1$ $y_2 = 2$ $y_3 = 9$

Switch polynomials:

$h_1(x) = (x-6)(x-7)/(5-6)(5-7) = \frac{1}{2}(x-6)(x-7)$

$h_2(x) = (x-5)(x-7)/(6-5)(6-7) = -(x-5)(x-7)$

$h_3(x) = (x-5)(x-6)/(7-5)(7-6) = \frac{1}{2}(x-5)(x-6)$

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

$= (6x^2 - 77x + 237)/2$

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

a) d+1 coefficients

$$a_d, a_{d-1}, \ldots, a_2, a_1, a_0$$

b) Its value at any d+1 points

$$P(x_1), P(x_2), \ldots, P(x_d), P(x_{d+1})$$

(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: $deg(P(x)) = d$

a) d+1 coefficients $a_d, a_{d-1}, \ldots, a_1, a_0$

b) Value at d+1 points $P(x_1), \ldots, P(x_{d+1})$

$P(0)$ . . $P(d)$

Adding two polynomials:

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

$P(0) = 5 \quad P(1) = 9 \quad P(2) = 6 \qquad (P+Q) = (8, 0, 11)$

$Q(0) = 3 \quad Q(1) = -9 \quad Q(2) = 5$

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) just requires 2(d+1) ~~additions~~ mults (if the two polynomials are already evaluated at the same points)

$P(0)=1$	$Q(0)=1$
$P(1)=2$	$Q(1)=9$
$P(2)=5$	$Q(2)=6$
$P(3)=2$	$Q(3)=1$
$P(4)=6$	$Q(5)=2$

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:

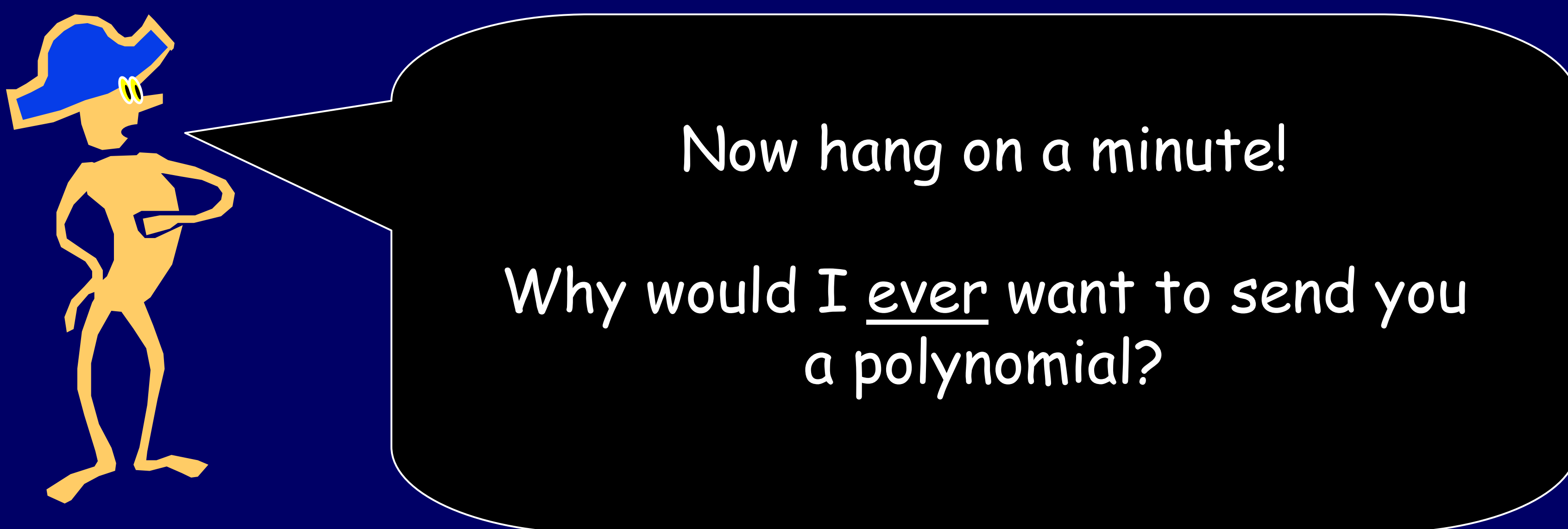
Is easy with representation (a)

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 corrupts my emails once in a while.



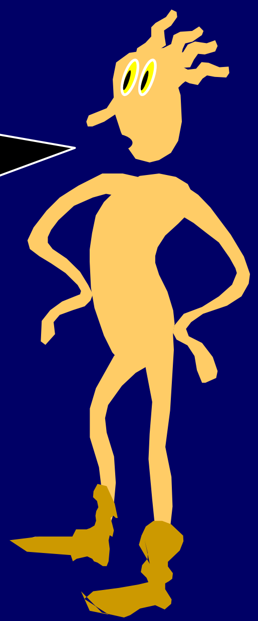
The value-representation is tolerant to "erasures"

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 send you a message
"hello"
I could write it as
"8 5 12 12 15"

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



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 (say) n > d+1 points and send
<k, P(k)>
to you for all k = 1, 2,…,d, …,n.

As long you get at least (d+1) of these,
choose any (d+1) of the ones you got, and reconstruct P(x).

But is it tolerant to “corruption” ?

I want to send you a polynomial P(x).

Suppose your mailer **corrupts** my emails once in a while.

E.g., suppose $P(x) = 2x^2 + 1$, and I chose n = 4.
I evaluated P(0) = 1, P(1) = 3, P(2) = 9, P(3) = 19.
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!

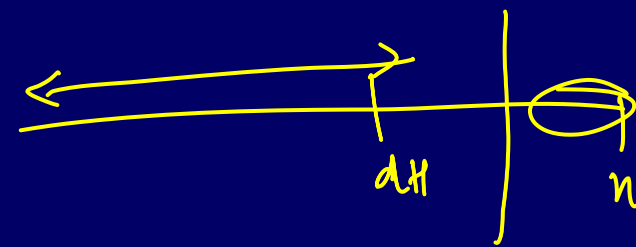
If we send the value of degree-d polynomial $P(x)$
at $n \geq d+1$ different points,

$\langle x_1, P(x_1)\rangle$, $\langle x_2, P(x_2)\rangle$, … , $\langle x_n, P(x_n)\rangle$

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

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

Also Error <u>**Correcting**</u> Representation



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

Ax = b

Error Correcting Codes (ECCs)

(We don't need to know which ones are corrupted.
Just that there are < (n-d)/2 corruptions.)

Berlekamp-
Welch
decoding

We can do this in class if we have enough time at the end…

And that's not all:
polynomials are amazing
in other ways as well…

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 ≤ 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 $Z_1, Z_2, \ldots, Z_n$ to the n officers such that:

a) if we have k of the Z_i’s, then we can find out S.

b) if we have k-1 Z_i’s, then any secret S is equally likely to have produced this set of Z_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 $Z_j = P(j)$

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 $Z_j = P(j)$

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

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

Since S is random, each such polynomial is equally likely to be chosen

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 $Z_j = P(j)$

If k officers get together, they can figure out P(x)

And then evaluate P(0) = S.

Adi Shamir

~~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 $Z_j = P(j)$

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!!!

Polynomials

Fundamental Theorem of polynomials:

Degree-d polynomial has at most d roots.

Two different deg-d polys agree on ≤ d points.

Lagrange Interpolation:

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

Gives us alternative representation for polys.

Many Applications of this representation

Error detecting/correcting codes

Secret sharing.