Diversification Improves Interpolation

Mark Giesbrecht **Daniel S. Roche**

WATERLOO
CHERITON SCHOOL OF COMPUTER SCIENCE
Symbolic Computation Group

ISSAC 2011
June 11, San Jose, California

Sparse Interpolation

The Problem

Given a **black box** for an unknown polynomial

$$f = c_1 x^{e_1} + c_2 x^{e_2} + \cdots + c_t x^{e_t},$$

determine the coefficients c_i and exponents e_i.

We are interested in two cases:

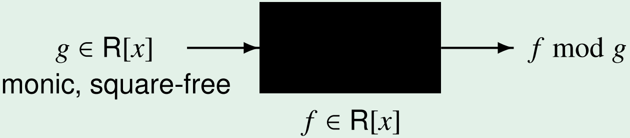
1. Coefficients come from a large, unchosen finite field.
2. Coefficients are approximations to complex numbers.

We first consider univariate interpolation over finite fields.

Remainder Black Box

We will use the following black box model
for univariate polynomials over a ring R:

The "Remainder Black Box"



The cost of the evaluation is $O(\mathsf{M}(\deg g))$.

This can be accomplished easily if f is given by an algebraic circuit, or by evaluating at roots of g (possibly over an extension of R).

Sparse interpolation algorithms over finite fields

Consider an unknown $f \in \mathbb{F}_q[x]$ with t terms and degree d.
Assume $q \gg d$ does not have any special properties.

- **Dense methods** (Newton/Waring/Lagrange): $\tilde{O}(d)$ total cost.
- **de Prony's method**
(Ben-Or & Tiwari '88, Kaltofen & Lakshman '89):
$O(t)$ probes; computation requires $O(t)$ discrete logarithms.
- **Garg & Schost '09**: $\tilde{O}(t^2 \log d)$ probes modulo degree-$\tilde{O}(t^2 \log d)$ polynomials; total cost $\tilde{O}(t^4 \log^2 d)$.
- **Ours**: $O(\log d)$ probes modulo degree-$\tilde{O}(t^2 \log d)$ polynomials; total cost $\tilde{O}(t^2 \log^2 d)$.

Garg & Schost's Algorithm

Consider (unknown) $f = c_1 x^{e_1} + c_2 x^{e_2} + \cdots + c_t x^{e_t}$.

Idea: Evaluate f mod $x^p - 1$ for a small prime p.
This gives $f_p = c_1 x^{e_1 \bmod p} + c_2 x^{e_2 \bmod p} + \cdots + c_t x^{e_t \bmod p}$.

If p is "good", then every $e_i \bmod p$ is distinct, and we have every coefficient and an unordered set $\{e_i \bmod p \mid 1 \leq i \leq t\}$.

Problem: How to correlate terms between different evaluations?

Garg & Schost's Algorithm

Consider (unknown) $f = c_1x^{e_1} + c_2x^{e_2} + \cdots + c_tx^{e_t}$.

Idea: Evaluate f mod $x^p - 1$ for a small prime p.
This gives $f_p = c_1x^{e_1 \bmod p} + c_2x^{e_2 \bmod p} + \cdots + c_tx^{e_t \bmod p}$.

If p is "good", then every $e_i \bmod p$ is distinct, and we have every coefficient and an unordered set $\{e_i \bmod p \mid 1 \le i \le t\}$.

Problem: How to correlate terms between different evaluations?

Consider the symmetric polynomial whose roots are the exponents: $\Gamma(z) = (z - e_1)(z - e_2)\cdots(z - e_t) \in \mathbb{Z}[z]$.

Coefficients of Γ have $\Theta(t \log d)$ bits, so we need this many "good prime" evaluations. Then we must find the integer roots of Γ.

Example 1 over R = $\mathbb{F}_{101}$

(**unknown**) $f = 49x^{42} + 46x^{30} + 7x^{27} \in \mathbb{F}_{101}[x]$

1. Evaluate $f(x)$ modulo $x^p - 1$ for small p:

$$f(x) \bmod (x^7 - 1) = 7x^6 + 46x^2 + 49$$
$$f(x) \bmod (x^{11} - 1) = 49x^9 + 46x^8 + 7x^5$$

Example 1 over R = $\mathbb{F}_{101}$

(**unknown**) $f = 49x^{42} + 46x^{30} + 7x^{27} \in \mathbb{F}_{101}[x]$

1. Evaluate $f(x)$ modulo $x^p - 1$ for small p:

$$f(x) \bmod (x^7 - 1) = 7x^6 + 46x^2 + 49$$
$$f(x) \bmod (x^{11} - 1) = 49x^9 + 46x^8 + 7x^5$$

2. Correlate terms using coefficients,
determine exponents with Chinese remaindering:

$$6 \bmod 7,\ 5 \bmod 11 \ \Rightarrow e_1 = 27$$
$$2 \bmod 7,\ 8 \bmod 11 \ \Rightarrow e_2 = 30$$
$$0 \bmod 7,\ 9 \bmod 11 \ \Rightarrow e_3 = 42$$

Example 2 over R = $\mathbb{F}_{101}$

(**unknown**) $f = 76x^{55} + 38x^{50} + 76x^{40} \in \mathbb{F}_{101}[x]$

1. Evaluate $f(x)$ modulo $x^p - 1$ for small p:

$$f(x) \bmod (x^7 - 1) = 76x^6 + 76x^5 + 38x^3$$
$$f(x) \bmod (x^{11} - 1) = 38x^8 + 76x^7 + 76$$

Example 2 over R = $\mathbb{F}_{101}$

(**unknown**) $f = 76x^{55} + 38x^{50} + 76x^{40} \in \mathbb{F}_{101}[x]$

1. Choose random $\alpha \in \mathbb{F}_{101}$: $\alpha = 18$
2. Evaluate $f(\alpha x)$ modulo $x^p - 1$ for small p:

$$f(\alpha x) \bmod (x^7 - 1) = 86x^6 + 47x^5 + 63x$$

$$f(\alpha x) \bmod (x^{11} - 1) = 47x^7 + 63x^6 + 86$$

Example 2 over R = $\mathbb{F}_{101}$

(**unknown**) $f = 76x^{55} + 38x^{50} + 76x^{40} \in \mathbb{F}_{101}[x]$

1. Choose random $\alpha \in \mathbb{F}_{101}$: $\alpha = 18$
2. Evaluate $f(\alpha x)$ modulo $x^p - 1$ for small p:

$$f(\alpha x) \bmod (x^7 - 1) = 86x^6 + 47x^5 + 63x$$
$$f(\alpha x) \bmod (x^{11} - 1) = 47x^7 + 63x^6 + 86$$

3. Correlate terms using coefficients,
   determine exponents with Chinese remaindering:

$$6 \bmod 7,\ 0 \bmod 11 \ \Rightarrow e_1 = 55$$
$$5 \bmod 7,\ 7 \bmod 11 \ \Rightarrow e_2 = 40$$
$$1 \bmod 7,\ 6 \bmod 11 \ \Rightarrow e_3 = 50$$

Example 2 over R = $\mathbb{F}_{101}$

(**unknown**) $f = 76x^{55} + 38x^{50} + 76x^{40} \in \mathbb{F}_{101}[x]$

1. Choose random $\alpha \in \mathbb{F}_{101}$: $\alpha = 18$
2. Evaluate $f(\alpha x)$ modulo $x^p - 1$ for small p:

$$f(\alpha x) \bmod (x^7 - 1) = 86x^6 + 47x^5 + 63x$$
$$f(\alpha x) \bmod (x^{11} - 1) = 47x^7 + 63x^6 + 86$$

3. Correlate terms using coefficients,
   determine exponents with Chinese remaindering:

$$6 \bmod 7,\ 0 \bmod 11 \ \Rightarrow e_1 = 55$$
$$5 \bmod 7,\ 7 \bmod 11 \ \Rightarrow e_2 = 40$$
$$1 \bmod 7,\ 6 \bmod 11 \ \Rightarrow e_3 = 50$$

4. Compute original coefficients of $f(x)$:

$$c_1 = 86/\alpha^{55} = 76, \qquad c_2 = 47/\alpha^{40} = 76, \qquad c_3 = 63/\alpha^{50} = 38$$

Diversification

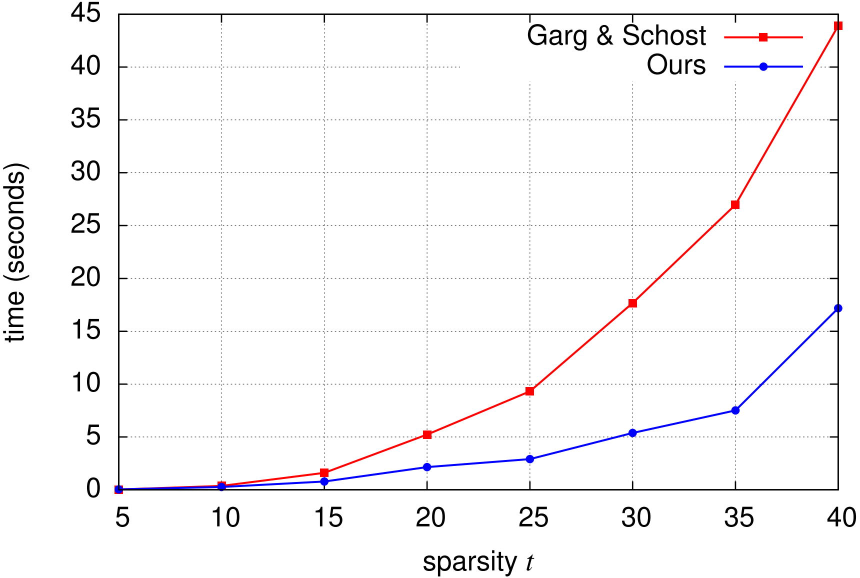
- We call a polynomial with all coefficients distinct diverse.
- Diverse polynomials are easier to interpolate.
- We use randomization to create diversity.

Theorem

If $f \in \mathbb{F}_q[x]$, $q \gg t^2 \deg f$, and $\alpha \in \mathbb{F}_q$ is chosen randomly, then $f(\alpha x)$ is diverse with probability at least $1/2$.

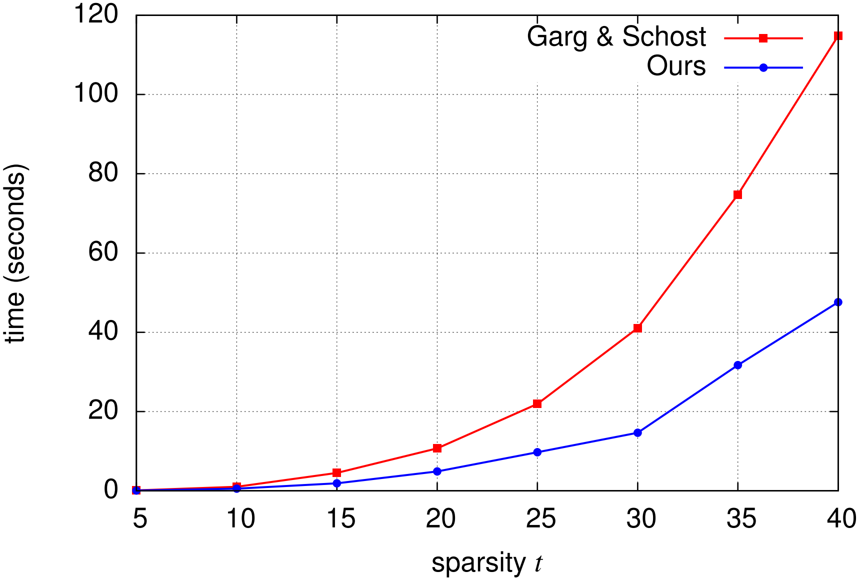
Interpolation over Finite Fields using Diversification

Degree $\approx 1\,000\,000$



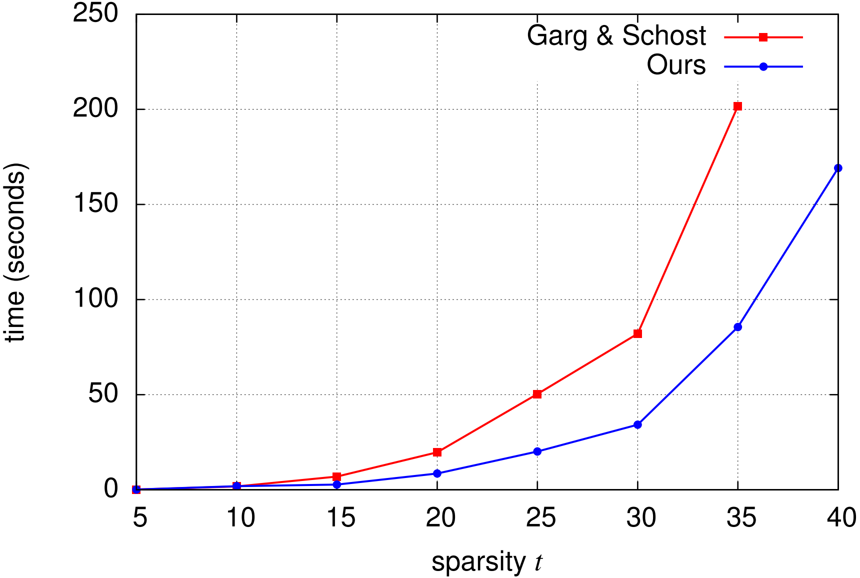
Interpolation over Finite Fields using Diversification

Degree $\approx 16\,000\,000$



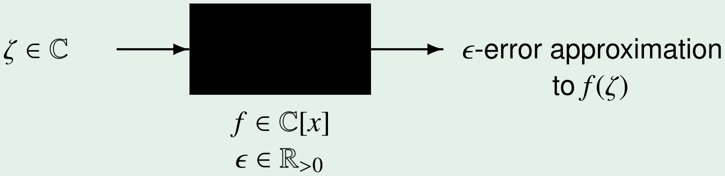
Interpolation over Finite Fields using Diversification

Degree $\approx 4\,000\,000\,000$



Approximate Sparse Interpolation over $\mathbb{C}[x]$

Approximate Black Box



- Related work: (G., Labahn, Lee ’06, ’09), (Kaltofen, Yang, Zhi ’07), (Cuyt & Lee ’08), (Kaltofen, Lee, Yang ’11).
- Applications to homotopy methods (e.g., Sommese, Verschelde, Wampler ’04).
- Known algorithms are fast but not provably stable.

Some numerical ingredients

We show that the sparse interpolation problem is well-posed for evaluations at low-order roots of unity:

Theorem

Suppose $f, g \in \mathbb{C}[x]$, p is a randomly-chosen "good prime", $\epsilon \in \mathbb{R}_{>0}$, and ω is a pth primitive root of unity.
If $\left|f(\omega^i) - g(\omega^i)\right| \le \epsilon \left|f(\omega^i)\right|$ for $0 \le i < p$, then $\|f - g\|_2 \le \epsilon\ \|f\|_2$.

- To use Garg & Schost's method, we need $f \bmod (x^p - 1)$.
- We compute $f(\exp(2j\pi\mathbf{i}/p))$ for $0 \le j < p$ and then use the FFT.
- The relative error on $f \bmod (x^p - 1)$ is the same as the relative error of each evaluation.

Example 3 over $\mathbb{C}$

(unknown)

$$f = (1.4 + 0.41\mathbf{i})\,x^{31} + (0.80 + 0.27\mathbf{i})\,x^{23} + (0.80 + 0.27\mathbf{i})\,x^{7} \in \mathbb{C}[x]$$

Example 3 over $\mathbb{C}$

(**unknown**)

$$f = (1.4 + 0.41\mathbf{i})\,x^{31} + (0.80 + 0.27\mathbf{i})\,x^{23} + (0.80 + 0.27\mathbf{i})\,x^{7} \in \mathbb{C}[x]$$

1. Choose $s \in \Omega(t^2) \Rightarrow s = 11$, random $k \in \{0, \ldots, s-1\} \Rightarrow k = 5$, then set $\alpha = \exp(\pi \mathbf{i} k/s)$

Example 3 over $\mathbb{C}$

(**unknown**)

$$f = (1.4 + 0.41\mathbf{i})\,x^{31} + (0.80 + 0.27\mathbf{i})\,x^{23} + (0.80 + 0.27\mathbf{i})\,x^{7} \in \mathbb{C}[x]$$

1. Choose $s \in \Omega(t^2) \Rightarrow s = 11$, random $k \in \{0, \ldots, s-1\} \Rightarrow k = 5$, then set $\alpha = \exp(\pi \mathbf{i} k/s)$
2. Evaluate $f(\alpha x)$ modulo $x^p - 1$ for small p using FFT:

$$\begin{aligned} f(\alpha x) \bmod (x^5 - 1) = {} & (0.00 + .01\mathbf{i}) + (.94 + 1.09\mathbf{i})x + (.083 + .84\mathbf{i})x^2 \\ & + (-.84 - .035\mathbf{i})x^3 + (0.01 + 0.00\mathbf{i})x^4 \end{aligned}$$

$$\begin{aligned} f(\alpha x) \bmod (x^7 - 1) = {} & (.085 + .84\mathbf{i}) + (-.01 + .003\mathbf{i})x + (-.84 - .035\mathbf{i})x^2 \\ & + (.94 + 1.08\mathbf{i})x^3 + (-.002 + .01\mathbf{i})x^4 \\ & + (.01 + 0.00\mathbf{i})x^5 + (0.00 - .002\mathbf{i})x^6 \end{aligned}$$

Example 3 over $\mathbb{C}$

(**unknown**)

$$f = (1.4 + 0.41\mathbf{i})\,x^{31} + (0.80 + 0.27\mathbf{i})\,x^{23} + (0.80 + 0.27\mathbf{i})\,x^{7} \in \mathbb{C}[x]$$

1. Choose $s \in \Omega(t^2) \Rightarrow s = 11$, random $k \in \{0, \ldots, s-1\} \Rightarrow k = 5$, then set $\alpha = \exp(\pi \mathbf{i} k/s)$
2. Evaluate $f(\alpha x)$ modulo $x^p - 1$ for small p using FFT:

$$f(\alpha x) \bmod (x^5 - 1) = (0.00 + .01\mathbf{i}) + (.94 + 1.09\mathbf{i})x + (.083 + .84\mathbf{i})x^2 + (-.84 - .035\mathbf{i})x^3 + (0.01 + 0.00\mathbf{i})x^4$$

$$f(\alpha x) \bmod (x^7 - 1) = (.085 + .84\mathbf{i}) + (-.01 + .003\mathbf{i})x + (-.84 - .035\mathbf{i})x^2 + (.94 + 1.08\mathbf{i})x^3 + (-.002 + .01\mathbf{i})x^4 + (.01 + 0.00\mathbf{i})x^5 + (0.00 - .002\mathbf{i})x^6$$

3. Correlate terms with close coefficients, determine exponents with Chinese remaindering
4. Compute original coefficients of $f(x)$

Approximate interpolation algorithm

Theorem

Let $f \in \mathbb{C}[x]$ with t terms and sufficiently large coefficients, $s \gg t^2$, and ω an s-PRU. Then for a random $k \in \{0, 1, \ldots, s-1\}$, $f(\omega^k x)$ has sufficiently separated coefficients (i.e., numerical diversity).

Cost: $\tilde{O}(t^2 \log^2 \deg f)$ evaluations at low-order roots of unity and floating point operations.

Experimental stability (degree 1 000 000, 50 nonzero terms):

Noise	Mean Error	Median Error	Max Error
0	4.440 e−16	4.402 e−16	8.003 e−16
$\pm 10^{-12}$	1.113 e−14	1.119 e−14	1.179 e−14
$\pm 10^{-9}$	1.149 e−11	1.191 e−11	1.248 e−11
$\pm 10^{-6}$	1.145 e−8	1.149 e−8	1.281 e−8

Extension to multivariate

Let $f \in \mathsf{R}[x_1, x_2, \ldots, x_n]$ with t terms and max degree $d - 1$.

Two techniques for extending a univariate sparse interpolation algorithm to multivariate (Kaltofen & Lee ’03):

Kronecker substitution. Create a black box for the univariate polynomial $\hat{f} = f(x, x^d, x^{d^2}, \ldots, x^{d^{n-1}})$, then interpolate $\hat{f}$.
Cost of our algorithm: $O(n^2 t^2 \log^2 d)$.

Zippel’s method. Go variable-by-variable; at each of n steps perform univariate interpolation t times on degree-d polynomials.
Cost of our algorithm: $O(nt^3 \log^2 d)$.

Future directions

Our algorithms perform more evaluations (probes) than $O(t)$, but do these at low-order roots of unity.

By randomized diversification, we avoid discrete logarithms and integer polynomial factorization.

Questions:

- Are discrete logarithms required to perform sparse interpolation using $O(t)$ evaluations over any finite field?
- Is there a trade-off between number of probes and computation cost/numerical stability?
- Can we weaken the diversification requirements (e.g., allow some collisions)?