Sparse Multivariate Interpolation

Alex Brandt

ORCCA, Department of Computer Science
University of Western Ontario

August 23, 2018

Sparse Multivariate Interpolation

- Interpolation is about finding a function which equals certain values at certain points. These points, are called the *interpolation nodes*.
- Interpolation is a fundamental technique for computer algebra, numerical analysis, engineering...
  - Can approximate very complex functions
  - Find a function for discrete data points
  - Evaluation-interpolation schemes in computer algebra
- We are interested in interpolating *sparse* functions in many variables.
  - ↳ Number of terms in a polynomial explode exponentially with increasing number of variables
- Convenient interpolants are polynomials and rational functions.
  - ↳ Easy to compute derivatives, integrals, evaluations, etc.

Plan

Introduction

Introduction: Polynomial Definitions

- **Variable** - a symbol representing some number.

  x, y, z, $\ldots$

- **Monomial** - a product of variables, each to some exponent.

  x^5yz^3

- **Coefficient** - a numerical multiplicative factor of a monomial.

  13, $7/9$, 2.1463

- **Polynomial** - a summation of coefficient-monomial products.

  $13x^5yz^3 + 7/9x^3y^2 + 11$

- **(Total) Degree** - the maximum sum of exponents of any monomial in a polynomial.

- **Partial Degree** - the maximum exponent of a particular variable in a polynomial.

Introduction: Sparse vs Dense Polynomials

Sparse and Dense have dual meanings for polynomials. A polynomial can *be* sparse or dense while also can be *represented* sparsely or densely.

- A polynomial *is sparse* if it has few non-zero coefficients. Conversely, a polynomial *is dense* if it has few zero coefficients.

  $x^9 + 1$ vs. $3x^4 + 7x^3 + 4x^2 + 1$

- A polynomial *is represented sparsely* if only its non-zero coefficients are stored while a polynomial *is represented densely* if all coefficients are stored.

  $1 \cdot x^9 + 1 \cdot x^0$ vs. $1 \cdot x^9 + 0 \cdot x^8 + 0 \cdot x^7 + \ldots + 1 \cdot x^0$

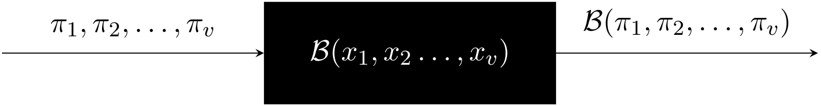
Introduction: Black-box functions

A *black-box* is some function, procedure, encoding, etc. of a mathematical function.

- Takes as input an evaluation point,
- Returns the value of the function at the input point.

The black-box is "opaque"

- One knows nothing of the underlying function
- Can only obtain evaluations at arbitrary points.



The Problem of Interpolation

The Problem of Interpolation

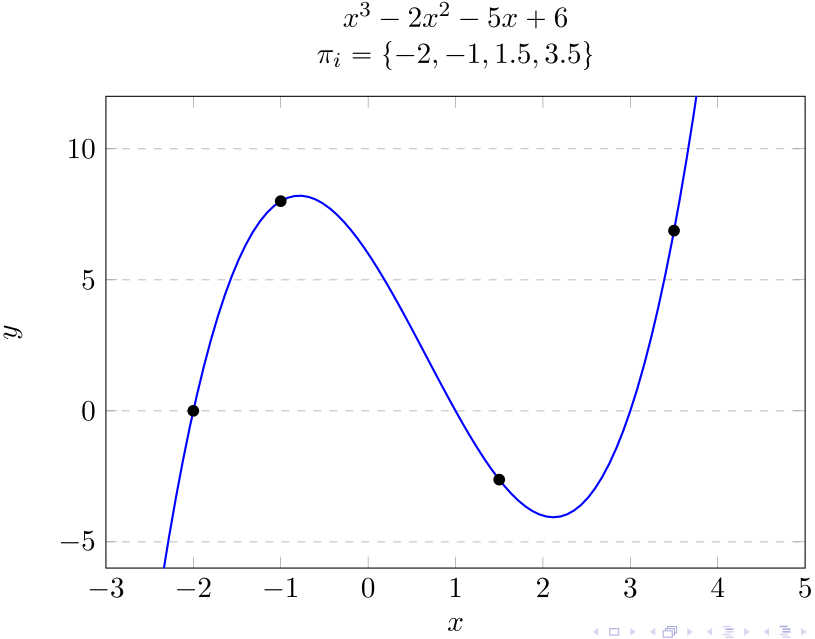
Problem 1: Given a set of point-value pairs, (π_i, β_i), $i = 1 \mathinner{..} n$, find a function f such that $f(\pi_i) = \beta_i$

Problem 2: Given a block-box encoding of a function, $\mathcal{B}$, find a function, f, such that $\mathcal{B}(\pi_i) = f(\pi_i)$ at sufficiently many π_i.

↳ If $\mathcal{B}$ is known to encode a polynomial (rational function) then the exact function can be recovered.

↳ Additional information is needed to define "sufficiently". Generally, the degree of the resulting interpolant.

The Problem of Interpolation



The Problem of Interpolation

Different "flavours" of interpolation exist

- Polynomial, Rational Function, Piece-wise Linear, etc.

One may define a function basis for the interpolation, a set of functions ϕ_j, $j = 1 .. m$, such that f is a linear combination of ϕ_j.

$$f = \alpha_1\phi_1(X) + \alpha_2\phi_2(X) + \cdots + \alpha_m\phi_m(X)$$

$$\implies \beta_i = \alpha_1\phi_1(\pi_i) + \alpha_2\phi_2(\pi_i) + \cdots + \alpha_m\phi_m(\pi_i)$$

The choice of function basis produces different flavours of interpolation. Choosing ϕ_j to be the set of monomials, we obtain *polynomial interpolation*.

- In the univariate case this set is $\{1, x, x^2, x^3, x^4, \dots\}$

The Problem of Interpolation

Given a basis of functions it is easy to set up a system of linear equations.

$$\beta_i = \alpha_1\phi_1(\pi_i) + \alpha_2\phi_2(\pi_i) + \cdots + \alpha_m\phi_m(\pi_i) \implies$$

$$\begin{bmatrix} \phi_1(\pi_1) & \phi_2(\pi_1) & \dots & \phi_m(\pi_1) \\ \phi_1(\pi_2) & \phi_2(\pi_2) & \dots & \phi_m(\pi_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\pi_n) & \phi_2(\pi_n) & \dots & \phi_m(\pi_n) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}$$

This matrix is called a *sample matrix*.

The Problem of Interpolation: Sparsity is a Necessity

The size of the multivariate monomial basis is exponential in the number of variables.

As the number of variables increases, it becomes prohibitively large to interpolate every coefficient for every monomial.

$$\{1,\ x,\ y,\ z,\ xy,\ xz,\ yz,\ xyz,\ x^2y,\ x^2z,\ xy^2,\ y^2z,\ xz^2,\ yz^2, x^3,\ y^3,\ z^3,\ \ldots\}$$

If the underlying function is sparse, then we want to take advantage of this structure to interpolate only the non-zero coefficients.

Univariate Interpolation

Univariate Interpolation

Univariate polynomial interpolation is the simplest. There is only one variable to determine.

Very straight-forward, direct solutions exist.

- Linear system solving
- Lagrange interpolation
- Newton interpolation (see [1, Chapter 13])

Univariate Interpolation: Linear system solving

Using the basis of functions: $\{\phi_1, \phi_2, \phi_3, \dots\} = \{1, x, x^2, \dots\}$ and point-value pairs (π_i, β_i) we get the system of equations:

$$\begin{bmatrix} 1 & \pi_1 & \pi_1^2 & \dots & \pi_1^{m-1} \\ 1 & \pi_2 & \pi_2^2 & \dots & \pi_2^{m-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \pi_n & \pi_n^2 & \dots & \pi_n^{m-1} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}$$

For $n = m$ this system can be solved to obtain a unique solution for $\alpha_1, \dots, \alpha_m$ and thus a unique interpolating polynomial.

- The sample matrix produced by univariate monomials is a *Vandermonde matrix*
- It is non-singular as long as π_i are pair-wise distinct

Univariate Rational Function Interpolation

Univariate rational functions can be interpolated by linear system solving with a simple modification.

$$R(x) = \frac{a(x)}{b(x)} \implies R(\pi_i) = \frac{a_1\pi_i^{d_1} + \cdots + a_n\pi_i^{d_n}}{b_1\pi_i^{e_1} + \cdots + b_m\pi_i^{e_m}} = \beta_i \implies$$

$$\begin{bmatrix} \pi_1^{d_1} & \dots & \pi_1^{d_n} & -\beta_1\pi_1^{e_1} & \dots & -\beta_1\pi_1^{e_m} \\ \pi_2^{d_1} & \dots & \pi_2^{d_n} & -\beta_2\pi_2^{e_1} & \dots & -\beta_2\pi_1^{e_m} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \pi_k^{d_1} & \dots & \pi_k^{d_n} & -\beta_k\pi_k^{e_1} & \dots & -\beta_k\pi_1^{e_m} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ b_1 \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \end{bmatrix}$$

This *homogeneous* system of equations either has the trivial solution or infinitely many solutions.

- ↳ *Normalize* one coefficient to 1 by multiplying $R(x)$ by $\varepsilon/\varepsilon = 1$
- ↳ Add a row to the system forcing one a_i or b_j to be 1
- ↳ Cannot guarantee a particular a_i or b_j is non-zero to be normalizeable, should try many.

Univariate Interpolation: Lagrange Interpolation

While linear system solving is possible, it is arithmetically expensive.

Lagrange Interpolation is more direct, providing an exact formula for the coefficients and basis functions of the interpolant.

$$f(x) = \sum_{j=1}^{n} \beta_j \phi_j(x) \qquad \phi_j(x) = \prod_{\substack{i=1 \\ i \neq j}}^{n} \frac{(x - \pi_i)}{(\pi_j - \pi_i)}$$

$$\phi_j(x) = \frac{(x - \pi_1) \dots (x - \pi_{j-1})(x - \pi_{j+1}) \dots (x - \pi_n)}{(\pi_j - \pi_1) \dots (\pi_j - \pi_{j-1})(\pi_j - \pi_{j+1}) \dots (\pi_j - \pi_n)}$$

Univariate Interpolation: Lagrange Interpolation

$$f(x) = \sum_{j=1}^{n} \beta_j \phi_j(x) \qquad \phi_j(x) = \prod_{\substack{i=1 \\ i \neq j}}^{n} \frac{(x-\pi_i)}{(\pi_j-\pi_i)}$$

- Coefficients are just function values, β_i
- Basis polynomial are designed so $\phi_j(\pi_j) = 1$, $\phi_j(\pi_i) = 0 \; i \neq j$
- Hence, $f(\pi_j) = \beta_j$ as required.

$$\begin{aligned} \phi_j(\pi_j) &= \frac{(\pi_j - \pi_1)\dots(\pi_j - \pi_{j-1})(\pi_j - \pi_{j+1})\dots(\pi_j - \pi_n)}{(\pi_j - \pi_1)\dots(\pi_j - \pi_{j-1})(\pi_j - \pi_{j+1})\dots(\pi_j - \pi_n)} \\ &= 1 \end{aligned}$$

$$\begin{aligned} f(\pi_j) &= \beta_1\phi_1(\pi_j) + \cdots + \beta_j\phi_j(\pi_j) + \cdots + \beta_n\phi_n(\pi_j) \\ &= \quad \beta_1 \cdot 0 \quad + \cdots + \quad \beta_j \cdot 1 \quad + \cdots + \quad \beta_n \cdot 0 \\ &= \quad \beta_j \end{aligned}$$

Univariate Interpolation: Lagrange Interpolation

Calculating the ϕ_j polynomials effectively is not immediately obvious.

For data locality while traversing the input data one can pre-compute many values and then never use the input data again.

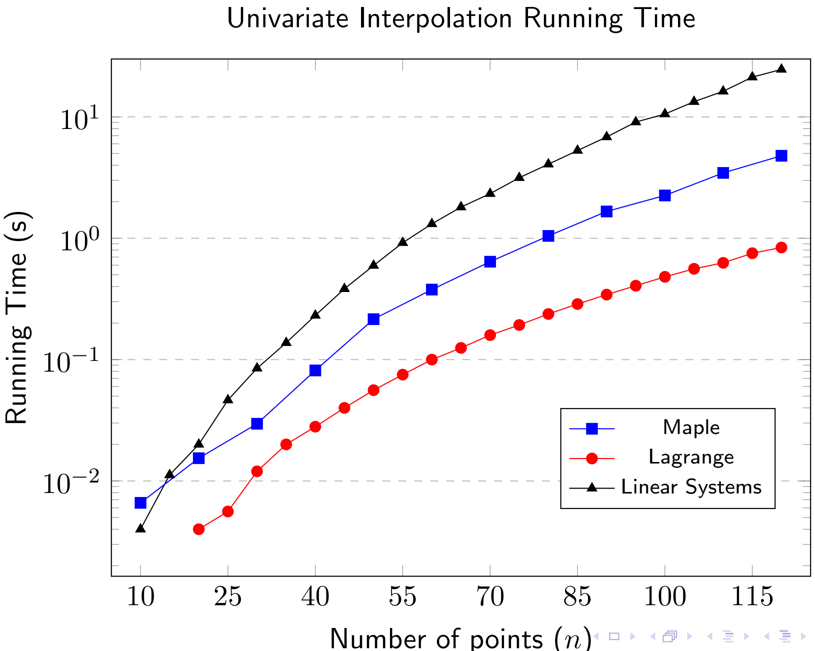
- Calculate modified coefficients $f_j = \beta_j \ / \ \prod_{i \neq j}(\pi_j - \pi_i)$,
- Generate factors $fact_i = (x - \pi_i)$ for $i = 1 \mathinner{..} n$

Expanding each ϕ_j can be effectively implemented by realizing one operand of each multiplication is always a monic binomial.

- Multiplication by a binomial can be implemented by a simultaneous shift, coefficient multiplication, and addition.
  - ↳ $a \cdot (x - c_i) = (ax - ac_i)$
  - ↳ ax is an increment of exponents of a, ac_i is only coefficient arithmetic.

Summation of ϕ_j can be done *in-place* to minimize memory allocation.

Univariate Interpolation: Experimentation



Sparse Multivariate Polynomial Interpolation

Sparse Polynomial Interpolation

The seminal work of Zippel [2] introduced a probabilistic method for sparse polynomial interpolation of a black-box. Zippel later extended this to be deterministic in [3].

Problem (Zippel's Sparse Polynomial Interpolation):
Given a black-box, $\mathcal{B}$, in v variables, and a *degree bound* for the interpolant, d, find a polynomial f such that they are equal at $t = \binom{v+d}{d}$ points.

$$\mathcal{B}(\pi_i) = f(\pi_i), \quad \text{for } i = 1 \mathinner{..} t$$

The maximum number of terms in a polynomial of v variables and degree d is $\binom{v+d}{d}$.

Sparse Polynomial Interpolation

Zippel's method involves interpolating each variable, one at a time.

This is accomplished by fixing higher-ordered variables at a specific *starting point* $\tilde{\pi} = (\zeta_1, \zeta_2, \ldots, \zeta_v)$

The structure (sparsity) of the polynomial is maintained at each step, interpolating only the non-zero polynomial terms.

$$
\begin{aligned}
f^{(0)} &= f(\zeta_1, \zeta_2, \ldots, \zeta_v) \in \mathbb{K}, \\
f^{(1)} &= f(x_1, \zeta_2, \ldots, \zeta_v) \in \mathbb{K}[x_1], \\
f^{(2)} &= f(x_1, x_2, \ldots, \zeta_v) \in \mathbb{K}[x_1, x_2], \\
&\vdots \\
f = f^{(v)} &= f(x_1, x_2, \ldots, x_v) \in \mathbb{K}[x_1, \ldots, x_v]
\end{aligned}
$$

Sparse Polynomial Interpolation

With the starting point $\tilde{\pi} = (\zeta_1, \zeta_2, \ldots, \zeta_v)$, a single variable can be interpolated by varying one value in the tuple.

Varying x_1, we can interpolate the points $\zeta_1, \omega_1, \ldots, \omega_d$ and values $\tilde{\beta}, \beta_1, \ldots, \beta_d$ to obtain $f^{(1)}$, say by Lagrange interpolation.

- $d + 1$ points to interpolate a maximum degree of d.

$$
\begin{aligned}
\mathcal{B}(\zeta_1, \zeta_2, \ldots, \zeta_v) &= \tilde{\beta}, \\
\mathcal{B}(\omega_1, \zeta_2, \ldots, \zeta_v) &= \beta_1, \\
\mathcal{B}(\omega_2, \zeta_2, \ldots, \zeta_v) &= \beta_2, \\
&\vdots \\
\mathcal{B}(\omega_d, \zeta_2, \ldots, \zeta_v) &= \beta_d
\end{aligned}
\qquad
\begin{aligned}
f^{(1)}(\zeta_1) &= \tilde{\beta}, \\
f^{(1)}(\omega_1) &= \beta_1, \\
f^{(1)}(\omega_2) &= \beta_2, \\
&\vdots \\
f^{(1)}(\omega_d) &= \beta_d
\end{aligned}
$$

Sparse Polynomial Interpolation

The result of the univariate interpolation has a special structure, each coefficient is really the evaluation of some function $g_i(x_2, \ldots, x_v)$ at $(\zeta_2, \ldots, \zeta_v)$.

$$\begin{aligned} f^{(1)} &= f(x_1, \zeta_2, \ldots, \zeta_v) \\ &= g_d(\zeta_2, \ldots, \zeta_v) x_1^d + g_{d-1}(\zeta_2, \ldots, \zeta_v) x_1^{d-1} + \cdots + g_0(\zeta_2, \ldots, \zeta_v) \\ &= \gamma_d \, x_1^d + \gamma_{d-1} \, x_1^{d-1} + \cdots + \gamma_0 \end{aligned}$$

There are $d+1$ different *coefficient functions*. Some of these may be identically zero while others may be zero at $(\zeta_2, \ldots, \zeta_v)$.

Coefficient functions which are zero are *assumed* to be zero, leading to a probabilistic method.

↳ Choosing ζ_i randomly from a larger set decreases probability that $g_i(\zeta_2, \ldots, \zeta_v)$ will evaluate to zero.

Sparse Polynomial Interpolation

Each **non-zero** g_i should now be interpolated for one variable, just as $f^{(1)}$ was for x_1.

- Requires obtaining $g_i(\nu_j, \zeta_3, \ldots, \zeta_n)$, $\nu_j \neq \zeta_2$, for $j = 1 \mathinner{..} d$

Let $g_i'(\nu_j) = g_i(\nu_j, \zeta_3, \ldots, \zeta_v)$, $\mathcal{B}'(x_1, \nu_j) = \mathcal{B}(x_1, \nu_j, \zeta_3, \ldots, \zeta_v)$

We can obtain $g_i'(\nu_j)$ as the solution to a linear system formed by evaluations of $\mathcal{B}'$. This makes use of the sparsity in $f^{(1)}$ compared to more univariate interpolations. Choosing ω_i randomly:

$$\begin{cases} \mathcal{B}'(\omega_1, \nu_j) = g_d'(\nu_j)\omega_1^d + g_{d-1}'(\nu_j)\omega_1^{d-1} + \cdots + g_0'(\nu_j) \\ \mathcal{B}'(\omega_2, \nu_j) = g_d'(\nu_j)\omega_2^d + g_{d-1}'(\nu_j)\omega_2^{d-1} + \cdots + g_0'(\nu_j) \\ \qquad \vdots \\ \mathcal{B}'(\omega_d, \nu_j) = g_d'(\nu_j)\omega_d^d + g_{d-1}'(\nu_j)\omega_d^{d-1} + \cdots + g_0'(\nu_j) \end{cases}$$

Sparse Polynomial Interpolation

Solving d systems of equations yields the values of each g_i at d distinct points.

Combined with the original $f^{(1)} = \gamma_d\, x_1^d + \gamma_{d-1}\, x_1^{d-1} + \cdots + \gamma_0$ we have $d+1$ points to interpolate a degree d polynomial in x_2.

$$
\begin{aligned}
f(x_1, \zeta_2, \ldots, \zeta_v) &= \gamma_d\, x_1^d + \gamma_{d-1}\, x_1^{d-1} + \cdots + \gamma_0 \\
f(x_1, \nu_1, \ldots, \zeta_v) &= g'_d(\nu_1)\, x_1^d + g'_{d-1}(\nu_1)\, x_1^{d-1} + \cdots + g'_0(\nu_1) \\
&\vdots \\
f(x_1, \nu_d, \ldots, \zeta_v) &= g'_d(\nu_d)\, x_1^d + g'_{d-1}(\nu_d)\, x_1^{d-1} + \cdots + g'_0(\nu_d)
\end{aligned}
$$

- Each g_i is interpolated as a function in x_2 by basic univariate interpolation using the points $\zeta_2, \nu_1, \ldots, \nu_d$ and values $\gamma_i, g'_1(\nu_1), \ldots, g'_d(\nu_d)$

Sparse Polynomial Interpolation

Lastly, each numerical coefficient γ_i in $f^{(1)}$ is replaced by $g_i(x_2)$ to obtain:

$$
\begin{aligned}
f^{(2)} &= f(x_1, x_2, \ldots, \zeta_v) \\
&= g'_d(x_2)\, x_1^d + g'_{d-1}(x_2)\, x_1^{d-1} + \cdots + g'_0(x_2) \\
&= \left(\gamma_{d,d} x_2^d + \cdots + \gamma_{d,0}\right) x^d + \cdots + \left(\gamma_{0,d} x_2^d + \cdots + \gamma_{0,0}\right)
\end{aligned}
$$

We repeat this process for each variable, adding one variable at each step:

(i) Solve linear systems to obtain evaluations of coefficient polynomials

(ii) Univariate interpolation of a single variable in each coefficient polynomial

Sparse Polynomial Interpolation: An Example

We look to interpolate $p = x^2y^2 + x^2yz + yz^2 + yz$ with degree bound $d = 4$. Written recursively:

$$p \in \mathbb{K}[y,z][x] = (y^2 + yz)x^2 + (yz^2 + yz)x^0$$
$$p \in \mathbb{K}[z][x,y] = (1)x^2y^2 + (z)x^2y + (z^2 + z)y$$

- Choose some starting point $\tilde{\pi} = (\zeta_x, \zeta_y, \zeta_z)$, $\mathcal{B}(\tilde{\pi}) = \tilde{\beta}$.
- $d = 4 \implies 5$ evaluations to interpolate each variable.

Stage 1: Interpolate $f^{(1)} \in \mathbb{K}[x]$

↳ Vary x: $\zeta_x, \omega_1, \omega_2, \omega_3, \omega_4$, while holding y and z constant at ζ_y and ζ_z. Let these points evaluate by $\mathcal{B}$ to $\tilde{\beta}, \beta_1, \beta_2, \beta_3, \beta_4$.

↳ Use $(\zeta_x, \tilde{\beta}), (\omega_1, \beta_1), (\omega_2, \beta_2), (\omega_3, \beta_3), (\omega_4, \beta_4)$ as input to some univariate interpolation, like Lagrange.

↳ Obtain $f^{(1)} = c_1x^2 + c_2$, $\ c_1, c_2 \in \mathbb{K}$.

Sparse Polynomial Interpolation: An Example

Stage 2: Interpolate $f^{(2)} \in \mathbb{K}[x, y]$

$$f^{(1)} = c_1 x^2 + c_2 \implies f^{(1)} = g_1(\zeta_y, \zeta_z) x^2 + g_2(\zeta_y, \zeta_z)$$

- Need 5 evaluations each of g_1 and g_2 to interpolate them.
- Vary y at the points $\zeta_y, \nu_1, \nu_2, \nu_3, \nu_4$, holding z fixed as ζ_z.
- For each new point, we solve a system of linear equations. Choose new, random, distinct points for x: ω_s, ω_t.

$$\begin{bmatrix} x^2|_{\omega_s} & x^0|_{\omega_s} \\ x^2|_{\omega_t} & x^0|_{\omega_t} \end{bmatrix} \begin{bmatrix} g_1(\nu_i, \zeta_z) \\ g_2(\nu_i, \zeta_z) \end{bmatrix} = \begin{bmatrix} \mathcal{B}(\omega_s, \nu_i, \zeta_z) \\ \mathcal{B}(\omega_t, \nu_i, \zeta_z) \end{bmatrix}$$

- Using the points ζ_y, ν_i and values $g_1(\zeta_y, \zeta_z)$, $g_1(\nu_i, \zeta_z)$, $i = 1..4$, interpolate $g_1(y, \zeta_z) = c_3 y^2 + c_4 y \in \mathbb{K}[y]$
- Similarly for g_2, obtaining $g_2(y, \zeta_z) = c_5 y$
- Expanding $f^{(1)}$, g_1, g_2 yields $f^{(2)} = c_3 x^2 y^2 + c_4 x^2 y + c_5 y$

Sparse Polynomial Interpolation: An Example

Stage 3: Interpolate $f = f^{(3)} \in \mathbb{K}[x, y, z]$

$$f^{(2)} = c_3 x^2 y^2 + c_4 x^2 y + c_5 y \implies f^{(2)} = g_3(\zeta_z) x^2 y^2 + g_4(\zeta_z) x^2 y + g_5(\zeta_z) y$$

- As before, we need 5 points. Reusing ζ_z we then need to solve 4 linear systems for the new points μ_1, μ_2, μ_3, μ_4.
- New random, distinct points for x: $\omega_s, \omega_t, \omega_u$ and y: ν_s, ν_t, ν_u.

$$\begin{bmatrix} x^2y^2|_{(\omega_s,\nu_s)} & x^2y|_{(\omega_s,\nu_s)} & y|_{(\omega_s,\nu_s)} \\ x^2y^2|_{(\omega_t,\nu_t)} & x^2y|_{(\omega_t,\nu_t)} & y|_{(\omega_t,\nu_t)} \\ x^2y^2|_{(\omega_u,\nu_u)} & x^2y|_{(\omega_u,\nu_u)} & y|_{(\omega_u,\nu_u)} \end{bmatrix} \begin{bmatrix} g_3(\mu_i) \\ g_4(\mu_i) \\ g_5(\mu_i) \end{bmatrix} = \begin{bmatrix} \mathcal{B}(\omega_s, \nu_s, \mu_i) \\ \mathcal{B}(\omega_t, \nu_t, \mu_i) \\ \mathcal{B}(\omega_u, \nu_u, \mu_i) \end{bmatrix}$$

- Using the points ζ_z, μ_i and values $g_3(\zeta_y)$, $g_3(\mu_i)$, $i = 1..4$, interpolate $g_3(z) = 1 \in \mathbb{K}[z]$
- Similarly for g_4 and g_5, obtaining $g_4(z) = z$, $g_5(z) = z^2 + z$
- Expanding $f^{(2)}$, g_3, g_4, g_5 yields

$$f^{(3)} = x^2y^2 + x^2yz + yz^2 + yz = f = p$$

Deterministic Sparse Polynomial Interpolation

The sources of error in the probabilistic methods are:

(i) Choice of starting point $\tilde{\pi} = (\zeta_1, \zeta_2, \ldots, \zeta_v)$ causing coefficient polynomials to evaluate to zero at $(\zeta_k, \ldots, \zeta_v)$

(ii) Singularity of matrix in each system of linear equations

Determinism: Choice of starting point $\tilde{\pi}$

$$f^{(1)} = g_d(\zeta_2, \ldots, \zeta_v)x_1^d + g_{d-1}(\zeta_2, \ldots, \zeta_v)x_1^{d-1} + \cdots + g_0(\zeta_2, \ldots, \zeta_v)$$

- Some $g_i(\zeta_2, \ldots, \zeta_v)$ maybe be identically zero while others may unluckily vanish at $(\zeta_2, \ldots, \zeta_v)$.

Proposition: For non-zero polynomials $g_1, \ldots, g_s \in \mathbb{K}[x_1, \ldots, x_v]$ where $\#(g_1) + \ldots + \#(g_s) = T$, if $\tilde{\pi}$ is a sequence of v different primes in $\mathbb{K}$, then $\exists j \in \mathbb{Z}$, $0 \leqslant j \leqslant T - s$, all of $g_i(\tilde{\pi}^j)$ are non-zero. [1, Proposition 102]

In a very brute-force way one can iterate through $T - s$ choices of starting point, choosing the one which yields the most non-zero g_i.

↳ $\#(g_i)$ is unknown beforehand. In implementation T is really a parameter which gives an upper bound on the number of terms.

$$\begin{cases} \mathcal{B}'(\omega_1,\nu_j) = g'_d(\nu_j)\omega_1^d + g'_{d-1}(\nu_j)\omega_1^{d-1} + \cdots + g'_0(\nu_j) \\ \mathcal{B}'(\omega_2,\nu_j) = g'_d(\nu_j)\omega_2^d + g'_{d-1}(\nu_j)\omega_2^{d-1} + \cdots + g'_0(\nu_j) \\ \qquad \vdots \\ \mathcal{B}'(\omega_d,\nu_j) = g'_d(\nu_j)\omega_d^d + g'_{d-1}(\nu_j)\omega_d^{d-1} + \cdots + g'_0(\nu_j) \end{cases}$$

When working on variable x_2, the polynomial is univariate and the matrix is a Vandermonde matrix in ω_i.

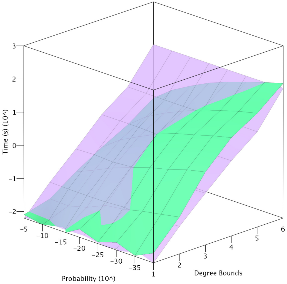
↳ It is non-singular as long as ω_i are distinct.

In later stages when interpolating x_k one must choose random points for all x_j, $j < k$. Say, $\vec{\omega}_i = (\omega_{i,1}, \omega_{i,2}, \ldots \omega_{i,k-1})$.

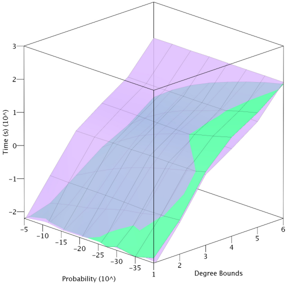
- Choosing $\vec{\omega}_i = (2^i, 3^i, 5^i, 7^i, \ldots)$, the first $k-1$ primes raised to the power i, ensures uniqueness when evaluating multivariate monomials and thus non-singularity [4].

Probabilistic vs Deterministic

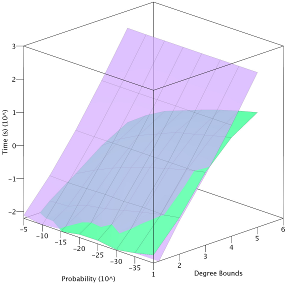
Running time for interpolating a sparse 3 variable function, f



$T = \#(f) + 3$



$T = \#(f) + 20$



$T = (d+1)^v$

- Green is probabilistic, purple is deterministic
- Probabilistic method varies with probability of getting incorrect result
- Both vary by *partial* degree bounds, d

Sparse Multivariate Rational Function Interpolation

Sparse Rational Function Interpolation (SRFI)

Problem (Sparse Rational Function Interpolation):
Given a black-box $\mathcal{B}$, encoding a rational function, $R(x_1, \ldots, x_v)$, and a *total degree bound* for the numerator and denominator, d and e, find the rational function R while being sensitive to its sparsity.

Cuyt and Lee propose a method for sparse rational function interpolation using a *homogenizing* variable [5].

- This method depends on both (dense) univariate and (sparse) multivariate interpolation.
- The use of sparsity in the rational function interpolation relies on the use of sparsity in the sparse multivariate interpolation.

SRFI: Homogenization

Given a rational function $R(x_1, \ldots, x_v) = \frac{a(x_1,\ldots,x_v)}{b(x_1,\ldots,x_v)}$ a new variable is introduced to produce an *auxiliary rational function*, $\tilde{R}$.

$$\tilde{R}(z, x_1, \ldots, x_v) = R(zx_1, \ldots, zx_v)$$

$$= \frac{A_0(x_1, \ldots, x_v) \cdot z^0 + A_1(x_1, \ldots, x_v) \cdot z^1 + \cdots + A_d(x_1, \ldots, x_v) \cdot z^d}{1 + B_1(x_1, \ldots, x_v) \cdot z^1 + \cdots + B_e(x_1, \ldots, x_v) \cdot z^e}$$

- The variable z groups together terms in a and b whose total degree is equal to the exponent on z.
- Assume $B_0(x_1, \ldots, x_v) = 1$
- $\tilde{R}$ can easily be interpolated as a univariate function in z

SRFI: Univariate Rational Interpolation

To interpolate z in $\tilde{R}(z, x_1, \ldots, x_v)$ fix $(x_1, \ldots, x_v)$ to be $(\zeta_1, \ldots, \zeta_v) = \vec{\zeta}$ and interpolate the univariate function $\tilde{R}(z, \zeta_1, \ldots, \zeta_v)$.

This requires $d + e + 1$ points for z. Use distinct ω_i as points and β_i as values, obtaining β_i from black-box evaluations:

$$\mathcal{B}(\omega_i \zeta_1, \ldots, \omega_i \zeta_v) = R(\omega_i \zeta_1, \ldots, \omega_i \zeta_v) = \tilde{R}(\omega_i, \zeta_1, \ldots, \zeta_v) = \beta_i$$

This yields a univariate function whose coefficients are the evaluations of the polynomials A_i and B_j at $(\zeta_1, \ldots, \zeta_v)$.

$$\tilde{R}(z, \vec{\zeta}) = \frac{A_0(\vec{\zeta}) \cdot z^0 + A_1(\vec{\zeta}) \cdot z^1 + \cdots + A_d(\vec{\zeta}) \cdot z^d}{1 + B_1(\vec{\zeta}) \cdot z^1 + \cdots + B_e(\vec{\zeta}) \cdot z^e}$$

SRFI: Multivariate Polynomial Interpolation

Univariate interpolation yields the evaluations of A_i and B_j. Many interpolations at various $\vec{\zeta}^{(k)}$ can obtain many evaluations for each A_i and B_j.

Using these evaluations, we can perform sparse multivariate interpolation on each A_i and B_j.

- ↳ The choice of multivariate interpolation scheme decides the values for $\vec{\zeta}^{(k)}$
- ↳ Example: $\vec{\zeta}^{(k)} = (2^k, 3^k, 5^k, 7^k, \dots)$ for deterministic Zippel

Notice each term in A_i has total degree i. One could create a specialized interpolation method since the degree of all terms are equal and known a priori.

SRFI: Algorithm

$$\tilde{R}(z,\vec{\zeta}) = \frac{A_0(\vec{\zeta})\cdot z^0 + A_1(\vec{\zeta})\cdot z^1 + \cdots + A_d(\vec{\zeta})\cdot z^d}{1 + B_1(\vec{\zeta})\cdot z^1 + \cdots + B_e(\vec{\zeta})\cdot z^e}$$

Sparse Rational Function Interpolation

Input: Block-box, $\mathcal{B}(x_1,\ldots,x_v)$, degree bounds, d and e

Output: $R(x_1,\ldots,x_v) = \frac{a_1 X^{\vec{d_1}} + \cdots + a_n X^{\vec{d_n}}}{1 + b_2 X^{\vec{e_2}} + \cdots + b_m X^{\vec{e_m}}}$

- For $k = 0, 1, \ldots$ until all A_i, B_j are interpolated
  - Decide $\vec{\zeta}^{(k)}$ based on sparse multivariate interpolation scheme
  - Using pairwise distinct $\omega_1,\ldots,\omega_{d+e+1}$, and evaluations $\mathcal{B}(\omega_j\zeta_1^{(k)},\ldots,\omega_j\zeta_v^{(k)})$ interpolate $\tilde{R}(z,\zeta_1^{(k)},\ldots,\zeta_v^{(k)})$
  - Add evaluations $A_i(\vec{\zeta^{(k)}})$ and $B_j(\vec{\zeta^{(k)}})$ to their respective ongoing multivariate interpolations and attempt to interpolate

SRFI: An Example

We look to interpolate the rational function $R(x, y)$ with a degree bound $d = e = 3$.

$$R(x, y) = \frac{x^2y + y^3 + x}{x^2 + y^2 + 1}$$

Conceptually, homogenization produces the auxiliary function $\tilde{R}(z, x, y)$:

$$\tilde{R}(z, x, y) = R(zx, zy) = \frac{(zx)^2(zy) + (zy)^3 + (zx)}{(zx)^2 + (zy)^2 + 1}$$

$$= \frac{z^3x^2y + z^3y^3 + zx}{z^2x^2 + z^2y^2 + 1}$$

$$= \frac{(x^2y + y^3)z^3 + (x)z}{(x^2 + y^2)z^2 + 1}$$

SRFI: An Example

$$\tilde{R}(z,x,y) = \frac{(x^2y+y^3)z^3+(x)z}{(x^2+y^2)z^2+1} \qquad = \frac{A_3(x,y)z^3+A_1(x,y)z}{B_2(x,y)z^2+1}$$

Use the probabilistic sparse polynomial interpolation method. For degree bound $d = e = 3$ each bi-variate polynomial coefficient needs *at most* 16 points to be interpolated.

- Generate $(\zeta_1^{(k)}, \zeta_2^{(k)})$ as needed by sparse interpolation for $k = 1\,..16$. The same points can be used for each A_i, B_j.
- For each k pick distinct $\omega_1, \ldots, \omega_{d+e+1=7}$
- Evaluate each $\mathcal{B}(\omega_i\zeta_1^{(k)}, \omega_i\zeta_2^{(k)})$ and interpolate a univariate rational function in z using $\omega_1, \ldots, \omega_7$

$$\tilde{R}(z,\zeta_1^{(k)},\zeta_2^{(k)}) = \frac{A_3(\zeta_1^{(k)},\zeta_2^{(k)})z^3 + A_1(\zeta_1^{(k)},\zeta_2^{(k)})z}{B_2(\zeta_1^{(k)},\zeta_2^{(k)})z^2+1}$$

SRFI: An Example

With each $\tilde{R}(z, \zeta_1^{(k)}, \zeta_2^{(k)})$ we gain evaluations of $A_3(\zeta_1^{(k)}, \zeta_1^{(k)})$, $A_1(\zeta_1^{(k)}, \zeta_1^{(k)})$, $B_2(\zeta_1^{(k)}, \zeta_1^{(k)})$.

- The very first univariate interpolation reveals this structure.
- Simultaneously A_3, A_1 and B_2 can be interpolated by the points $(\zeta_1^{(k)}, \zeta_2^{(k)})$ and their respective evaluations.

Sparse interpolations yield $A_3(x, y) = x^2 y + y^3$, $A_1(x, y) = x$, $B_2 = x^2 + y^2$. Simply ignore z and combine A_i to form numerator and B_j for denominator.

$$R(x, y) = \frac{A_3(x, y) + A_1(x, y)}{B_2(x, y) + 1} = \frac{x^2 y + y^3 + x}{x^2 + y^2 + 1}$$

SRFI: Shifted Basis

The previous discussion assumed that the denominator did not vanish at $(0, \ldots, 0)$, that is $b_0 X^{\vec{0}} = 1$

↳ Guarantees ability to normalize the rational function

↳ In general, one cannot guarantee *any* particular term is non-zero, so normalization cannot occur.

Using a shifted power basis, we can instead force the normalization of the auxiliary function.

Let $\vec{\sigma} = (\sigma_1, \ldots, \sigma_v)$ be a point at which $R(\vec{\sigma})$ is defined.

$$\begin{aligned}
\hat{R}(z, x_1, \ldots, x_v) &= R(zx_1 + \sigma_1, zx_2 + \sigma_2, \ldots, zx_v + \sigma_v) \\
&= \frac{\hat{a}(z)}{\hat{b}(z)} \\
&= \frac{\hat{A}_0(x_1, \ldots, x_v)z^0 + \cdots + \hat{A}_d(x_1, \ldots, x_v)z^d}{\hat{B}_0(x_1, \ldots, x_v)z^0 + \cdots + \hat{B}_e(x_1, \ldots, x_v)z^e}
\end{aligned}$$

SRFI: Shifted Basis

$$b = \sum_{j=1}^{m} b_j x_1^{e_{j,1}} \dots x_v^{e_{j,v}} \implies \hat{b}(z) = b(zx_1 + \sigma_1, \dots, zx_v + \sigma_v)$$

$$= \sum_{j=1}^{m} b_j (zx_1 + \sigma_1)^{e_{j,1}} \dots (zx_v + \sigma_v)^{e_{j,v}}$$

Evaluating $\hat{b}$ at $z = 0$ yields the constant term, $\hat{B}_0 \neq 0$.

↳ $\hat{R}$ can be normalized, forcing $\hat{B}_0(x_1, \dots, x_v) = 1$

↳ With normalization the univariate interpolation can occur

$$\begin{aligned}\hat{b}(0) &= \hat{B}_0(x_1, \dots, x_v) \\ &= \sum_{j=1}^{m} b_j (\sigma_1)^{e_{j,1}} \dots (\sigma_v)^{e_{j,v}} \\ &= b(\sigma_1, \dots, \sigma_v) \neq 0\end{aligned}$$

SRFI: Shifted Basis

$$\begin{aligned}\hat{R}(z, x_1, \ldots, x_v) &= R(zx_1 + \sigma_1, zx_2 + \sigma_2, \ldots, zx_v + \sigma_v) \\ &= \frac{\hat{A}_0(x_1, \ldots, x_v)z^0 + \cdots + \hat{A}_d(x_1, \ldots, x_v)z^d}{1 + \hat{B}_1(x_1, \ldots, x_v)z^1 + \cdots + \hat{B}_e(x_1, \ldots, x_v)z^e}\end{aligned}$$

Finally, one can interpolate each $\hat{A}_i$ and $\hat{B}_j$ as before.

- ↳ With the shift, each $\hat{A}_i$ and $\hat{B}_j$ has densified.
- ↳ Once the shift is removed, many terms will become zero and the sparsity recovered.
- ↳ More advanced schemes exist to recover sparsity while in the shifted basis.

Future Work

Sparse interpolation schemes lend themselves to parallelization.

- Multivariate Polynomial Interpolation
  - ↳ Each coefficient polynomial g_i can be interpolated in parallel.
  - ↳ For the deterministic variation, the interpolation for each choice of starting point can be run in parallel.
  - ↳ With many threads, it is likely that the deterministic algorithm could surpass the probabilistic.
- Multivariate Rational Function Interpolation
  - ↳ The coefficient polynomials A_i, B_j can all be interpolated in parallel.
  - ↳ Current experimentation focused solely on number of black-box evaluations, look to view actual running time.

References

[1] R. Zippel. *Effective polynomial computation*. Vol. 241. Springer Science & Business Media, 2012.

[2] R. Zippel. "Probabilistic algorithms for sparse polynomials". In: *Symbolic and algebraic computation*. Springer, 1979, pp. 216–226.

[3] R. Zippel. "Interpolating polynomials from their values". In: *Journal of Symbolic Computation* 9.3 (1990), pp. 375–403.

[4] M. Ben-Or and P. Tiwari. "A deterministic algorithm for sparse multivariate polynomial interpolation". In: *Proceedings of the twentieth annual ACM symposium on Theory of computing*. ACM. 1988, pp. 301–309.

[5] A. Cuyt and W.-s. Lee. "Sparse interpolation of multivariate rational functions". In: *Theoretical Computer Science* 412.16 (2011), pp. 1445–1456.

Thank you!

Questions?