Symbolic-Numeric Sparse Interpolation of Multivariate Polynomials

Mark Giesbrecht **George Labahn** **Wen-shin Lee**
Symbolic Computation Group
University of Waterloo

April 5, 2003

Sparse interpolation of a black box polynomial

Black box $f = \sum_{j=1}^{t} c_j x_1^{d_{j,1}} \cdots x_n^{d_{j,n}}$

$$p_1, \ldots, p_n \longrightarrow \blacksquare \longrightarrow f(p_1, \ldots, p_n)$$

Sparse Interpolation

$$\tilde{f} = \sum_{j=1}^{t} \tilde{c}_j x_1^{d_{j,1}} \cdots x_n^{d_{j,n}}$$

Determine $\tilde{c}_j$:

1. exactly
2. approximately, to a fixed precision

Example Black box $10x^6y^8 - 6x^{10} - 5x^8y - 4y^7$

	Exact	Approximate
Input	$\omega_x = \exp(\frac{2\pi i}{31})$ (31st-PRU)	$\mathrm{evalf}(\omega_x) = 0.9795299413 + 0.2012985201I$
	$\omega_y = \exp(\frac{2\pi i}{37})$ (37th-PRU)	$\mathrm{evalf}(\omega_y) = 0.9856159104 + 0.1690008203I$
Compute	$f(\omega_x^i, \omega_y^i)$	$\mathrm{evalf}(f(\mathrm{evalf}(\omega_x^i), \mathrm{evalf}(\omega_y^i))), i = 0, 1, \ldots, 8$
Output	$10x^6y^8$	$(10.00000006 - 0.8543610430 \times 10^{-8}I)x^6y^8$
	$-6x^{10}$	$(-6.000000235 - 0.1390185436 \times 10^{-6}I)x^{10}$
	$-5x^8y$	$(-4.999999825 + 0.1968676105 \times 10^{-6}I)x^8y$
	$-4y^7$	$(-3.999999997 - 0.493054565410^{-7}I)y^7$

Methods: Prony (1795) $\sim$ Ben-Or/Tiwari (1988)

A sum of exponential functions $F(x) = \sum_{j=1}^{t} c_j e^{\mu_j x} = \sum_{j=1}^{t} c_j b_j^x$	A polynomial $f(x_1,\ldots,x_n) = \sum_{j=1}^{t} c_j x_1^{d_{j,1}} \cdots x_n^{d_{j,n}}$
1. Solve λ_j, $i = 0,\ldots,t-1$: $\sum_{j=0}^{t-1} \lambda_j F(i+j) = -F(i+t)$ 2. $e^{\mu_j} = b_j$ are zeros of $\Lambda = z^t + \lambda_{t-1} z^{t-1} + \cdots + \lambda_0$	1. Compute† the minimal Λ that generates* $\{f(p_1^i,\ldots,p_n^i)\}_{i=0}^{2t-1}$ 2. $p_1^{d_{j,1}} \cdots p_n^{d_{j,n}}$ are zeros of $\Lambda = z^t + \lambda_{t-1} z^{t-1} + \cdots + \lambda_0$
3. Determine c_j from $e^{\mu_i} = b_j$ and evaluations of F	3. Determine c_j from $p_1^{d_{j,1}} \cdots p_n^{d_{j,n}}$ and evaluations of f

† Berlekamp/Massey algorithm

$*$ $p_1,\ldots,p_n$ relatively prime

Numerical challenges in Prony's method

Ill-conditioned Hankel system

$$\underbrace{\begin{bmatrix} F(0) & F(1) & \dots & F(t-1) \\ F(1) & F(2) & \dots & F(t) \\ \vdots & \vdots & \ddots & \vdots \\ F(t-1) & F(t) & \dots & F(2t-2) \end{bmatrix}}_{H_{0,t-1}} \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_{t-1} \end{bmatrix} = - \begin{bmatrix} F(t) \\ F(t+1) \\ \vdots \\ F(2t-1) \end{bmatrix}$$

Root-finding sensitive to perturbations in λ_j

$$\Lambda = z^t + \lambda_{t-1} z^{t-1} + \cdots + \lambda_0 = 0$$

Further challenge in Ben-Or/Tiwari algorithm

Recover multivariate terms in the target polynomial

Generalized eigenvalue reformulation (Golub, Milanfar, and Varah 1999)

$$H_{0,t-1} = \underbrace{\begin{bmatrix} 1 & \dots & 1 \\ b_1 & \dots & b_t \\ \vdots & \vdots & \vdots \\ b_1^{t-1} & \dots & b_t^{t-1} \end{bmatrix}}_{V} \underbrace{\begin{bmatrix} c_1 & 0 & \dots & 0 \\ 0 & c_2 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & c_t \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} 1 & b_1 & \dots & b_1^{t-1} \\ 1 & b_2 & \dots & b_2^{t-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & b_t & \dots & b_t^{t-1} \end{bmatrix}}_{V^T}$$

$$\underbrace{\begin{bmatrix} F(1) & \dots & F(t) \\ \vdots & \ddots & \vdots \\ F(t) & \dots & F(2t-1) \end{bmatrix}}_{H_{1,t}} = VDBV^T \quad \text{with } B = \begin{bmatrix} b_1 & 0 & \dots & 0 \\ 0 & b_2 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ddots & 0 & b_t \end{bmatrix}$$

$$V^{-1}H_{0,t-1}V^{-T} = D, \qquad V^{-1}H_{1,t}V^{-T} = DB$$

$\Longrightarrow H_{1,t}v = bH_{0,t-1}v$ has solutions $b_1, \dots, b_t$ for b.

Univariate sparse interpolation via generalized eigenvalues

$$f(x) = \sum_{j=1}^{t} c_j x^{d_j}$$

$$\underbrace{\begin{bmatrix} f(p^0) & f(p) & \dots & f(p^{t-1}) \\ f(p) & f(p^2) & \dots & f(p^t) \\ \vdots & \vdots & \ddots & \vdots \\ f(p^{t-1}) & f(p^t) & \dots & f(p^{2t-2}) \end{bmatrix}}_{H_{0,t-1}} v = z \underbrace{\begin{bmatrix} f(p) & f(p^{t+1}) & \dots & f(p^t) \\ f(p^2) & f(p^3) & \dots & f(p^{t+1}) \\ \vdots & \vdots & \ddots & \vdots \\ f(p^t) & f(p^{t+1}) & \dots & f(p^{2t-1}) \end{bmatrix}}_{H_{1,t}} v$$

- Solutions for $z : \tilde{b}_1, \tilde{b}_2, \dots, \tilde{b}_t$ approximate $p^{d_1}, p^{d_2}, \dots, p^{d_t}$.
- Obtain candidates for $d_1, d_2, \dots, d_t$ from p and $\tilde{b}_1, \tilde{b}_2, \dots, \tilde{b}_t$. (There can be more than t candidates.)
- Compute $\tilde{c}_j$ from a transpose Vandermonde system that is based on terms with exponents the candidates for $d_1, d_2, \dots, d_t$. (For $H_{0,t-1}$ and $H_{1,t}$, the system can be as large as $2t \times 2t$.)

Multivariate case

$$f(x_1,\ldots,x_n) = \sum_{j=1}^{t} c_j x_1^{d_{j,1}} \cdots x_n^{d_{j,n}}$$

Variable by variable (sometimes called “peeling method.”)

- Interpolate one variable at a time via univariate interpolation.
- Numerically suspect?

Everything at once (numerically better)

- Evaluate each variable at powers of a primitive root of unity of orders that are relatively prime.

$$\omega_k^i = \exp(2\pi i/p_k) \text{ and } f(\omega_1^i, \ldots, \omega_n^i)$$
$$p_1, \ldots, p_n \text{ relatively prime.}$$

Recall: In the exact arithmetic, the original Ben-Or/Tiwari algorithm evaluate $f(p_1^i, \ldots, p_n^i)$ for $p_1, \ldots, p_n$ relatively prime.

Require the number of terms (or an upper bound)

Binary search

Guess an upper bound $\tau \geq t$, double τ if fails.

Early termination heuristic

Cabay-Meleshko algorithm: a fast procedure estimates the condition number of a Hankel matrix $H_{0,N}$ for any N.

Simultaneous diagonalizations

The general eigenvalue reformulation can be applied to interpolation systems M, N containing:

$$F^{-1}MG^{-1} = \bar{D}, F^{-1}NG^{-1} = \bar{D}\bar{B} \text{ with } \bar{D}, \bar{B} \text{ diagonal.}$$

Sparse interpolation in the Chebyshev basis

$$f(x) = \sum_{j=1}^{t} c_j T_{d_j}(x) \text{ with Chebyshev basis } T_k(x)$$

$$a_i = f(T_i(p))$$

$$HT = \begin{bmatrix} 2a_0 & 2a_1 & \dots & 2a_{t-1} \\ 2a_1 & a_2 + a_0 & \dots & a_t + a_{t-2} \\ \vdots & \vdots & \ddots & \vdots \\ 2a_{t-1} & a_t + a_{t-2} & \dots & a_{2t-2} + a_0 \end{bmatrix}$$

$$HT_{\uparrow} = \begin{bmatrix} 2a_1 & a_2 + a_0 & \dots & a_t + a_{t-2} \\ 2a_2 & a_3 + a_1 & \dots & a_{t+1} + a_{t-3} \\ \vdots & \vdots & \ddots & \vdots \\ 2a_t & a_{t+1} + a_{t-1} & \dots & a_{2t-1} + a_1 \end{bmatrix}$$

$$HT_{\downarrow} = \begin{bmatrix} 2a_1 & a_2+a_0 & \dots & a_t+a_{t-2} \\ 2a_0 & 2a_1 & \dots & 2a_{t-1} \\ 2a_1 & a_2+a_0 & \dots & a_t+a_{t-2} \\ \vdots & \vdots & & \vdots \\ 2a_{t-2} & a_{t-1}+a_{t-3} & \dots & a_{2t-3}+a_1 \end{bmatrix}$$

$$\boxed{\tfrac{1}{2}(HT_{\uparrow}+HT_{\downarrow})v = zHTv}$$

- $T_{d_1}(p), \dots, T_{d_t}(p)$ are solutions for z.
- For $-1 \le x \le 1$, $T_n(x) = \cos n\theta$ for $x = \cos\theta$.

  $\Longrightarrow T_n(x) = \cos n(\arccos x)$ where $0 \le \arccos x \le \pi$

Sparse interpolation in factorial bases

$$f(x) = \sum_{j=1}^{t} c_j x^{\bar{d}_j}, \qquad x^{\bar{n}} = x(x+1)\cdots(x+n-1)$$

$$\Delta(f(x)) = f(x+1) - f(x)$$

$$f^{(i)}(x) = x\Delta(f^{(i-1)}(x)) = \sum_{j=1}^{t} d_j^i c_j x^{\bar{d}_j}, \qquad p > 0$$

$$\underbrace{\begin{bmatrix} f^{(0)}(p) & \dots & f^{(t-1)}(p) \\ \vdots & \ddots & \vdots \\ f^{(t-1)}(p) & \dots & f^{(2t-2)}(p) \end{bmatrix}}_{H_{0,t-1}} \qquad \underbrace{\begin{bmatrix} f^{(1)}(p) & \dots & f^{(t)}(p) \\ \vdots & \ddots & \vdots \\ f^{(t)}(p) & \dots & f^{(2t-1)}(p) \end{bmatrix}}_{H_{1,t}}$$

$\boxed{H_{1,t}v = zH_{0,t-1}v}$ $\quad d_1, \dots, d_t$ are solutions for z.

- The case for the falling factorials can be derived similarly.
- The difference operator Δ behaves in the same way as a derivative.