Symbolic-Numeric Sparse Interpolation of Multivariate Polynomials

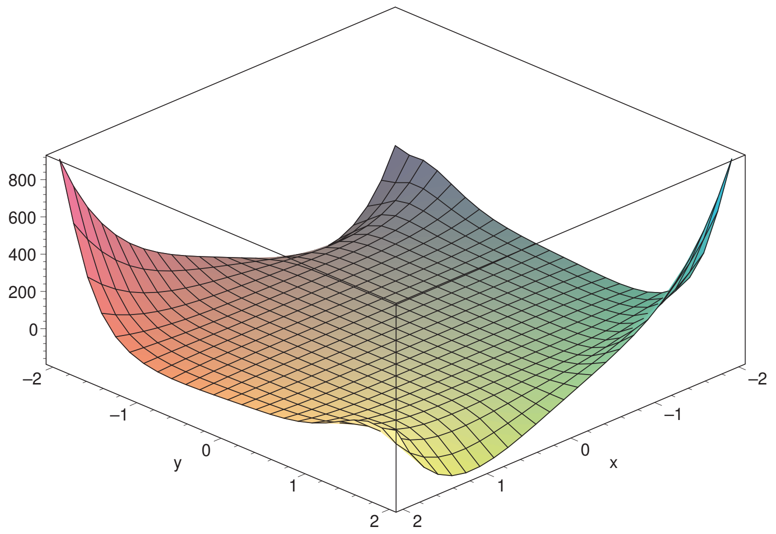
University of Waterloo

ORCCA
Ontario Research Centre for Computer Algebra

Mark Giesbrecht **George Labahn** **Wen-shin Lee**
School of Computer Science
University of Waterloo

Approximate interpolation of polynomials

Suppose we can sample the value of a multivariate polynomial



How do we recover the explicit representation?

$$f(x,y) = x^5y + 0.1xy^{13} - 0.5xy^{11} + 2.2x^4y^4$$

What if $f(x_1, \ldots, x_n)$ is sparse?

Approximate interpolation of polynomials (cont.)

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

In exact arithmetic:

Zippel's probabilistic interpolation (1979)

variable by variable

Ben-Or/Tiwari deterministic algorithm (1988)

all variables at once

What about floating point arithmetic?

Ben-Or/Tiwari algorithm: $f(x,y) = x^5y + 0.1xy^{13} - 0.5xy^{11} + 2.2x^4y^4$

- Pick p_1, p_2 relatively prime: say $p_1 = 2,\ p_2 = 3$.
  Evaluate $a_0 = f(2^0, 3^0), a_1 = f(2,3), a_2 = f(2^2, 3^2), \ldots$.

  sequence $a_0, a_1, a_2, \ldots$ is linearly generated

- Compute the minimal generator $G(z)$ by Berlekamp/Massey algorithm or solving a Hankel system.

$$\begin{aligned} G(z) = {} & z^4 - 3544332z^3 - 1134650042820z^2 \\ & - 1573008457549248z - 1405550128432803844 \end{aligned}$$

- Roots of $G(z) = 0$ are non-zero terms in f at $x = p_1,\ y = p_2$:

$$\begin{array}{cccc} 96 = 2^5 \cdot 3, & 1296 = 2^4 \cdot 3^4, & 354294 = 2 \cdot 3^{11}, & 3188646 = 2 \cdot 3^{13} \\ \Downarrow & \Downarrow & \Downarrow & \Downarrow \\ x^5y & x^4y^4 & xy^{11} & xy^{13} \end{array}$$

- Recover coefficients by solving a Vandermonde system.

Gaspard Clair Franois Marie Riche de Prony

Essai expérimental et analytique sur les lois de la dilatabilité et sur celles de la force expansive de la vapeur de l'eau et de la vapeur de l'alkool, à diff'erentes températures.
J. de l' École Polytechnique 1:24–76, 1795.

For a function $F : \mathbb{R} \to \mathbb{R}$, and $t \in \mathbb{Z}_{>0}$, find F_j, δ_j such that

$$F(x) = \sum_{j=1}^{t} F_j e^{\delta_j x}$$

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

A sum of exponential functions $F(x) = \sum_{j=1}^{t} c_j e^{\delta_j x}$	A polynomial $f(x_1, \ldots, x_n) = \sum_{j=1}^{t} c_j x_1^{d_{1,j}} \cdots x_n^{d_{n,j}}$
1. Solve g_j, $i = 0, \ldots, t-1$: $\sum_{j=0}^{t-1} g_j F(i+j) = -F(i+t)$ Evaluate $F(0), F(1), F(2), \ldots$. 2. e^{δ_j} are zeros of $G(z) = z^t + g_{t-1} z^{t-1} + \cdots + g_0$	Pick $p_1, \ldots, p_n$ relatively prime 1. Compute the minimal G that generates $\{f(p_1^i, \ldots, p_n^i)\}_{i=0}^{2t-1}$ Evaluate $f(p^0), f(p^1), f(p^2), \ldots$. 2. $p_1^{d_{1,j}} \cdots p_n^{d_{n,j}}$ are zeros of $G(z) = z^t + g_{t-1} z^{t-1} + \cdots + g_0$
3. Determine c_j from e^{δ_i} and evaluations of F	3. Determine c_j from $p_1^{d_{1,j}} \cdots p_n^{d_{n,j}}$ and evaluations of f

Numerical challenges in Prony's method

When finding the generating polynomial...

$$G(z) = z^t + g_{t-1}z^{t-1} + \cdots + g_0$$

done via solving $\underbrace{\begin{bmatrix} F(0) & F(1) & \dots & F(t-1) \\ F(1) & F(2) & \dots & F(t) \\ \vdots & \vdots & \ddots & \vdots \\ F(t-1) & F(t) & \dots & F(2t-2) \end{bmatrix}}_{H_0} \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{t-1} \end{bmatrix} = - \begin{bmatrix} F(t) \\ F(t+1) \\ \vdots \\ F(2t-1) \end{bmatrix}$

- ☹ Hankel system H_0 is often ill-conditioned...
- ☹ Root-finding is not a good thing to do numerically: very sensitive to perturbations in g_j.
- ☹ To recover coefficients c_j in $F(x) = \sum_{j=1}^{t} c_j e^{\delta_j x}$, need to solve a Vandermonde system that might be ill-conditioned.

Numerical challenges in Prony's method (2)

Consider univariate $f(x) = \sum_{1\le j\le t} f_j x^{d_j}, \quad b_j = p^{d_j}$

☹ $\|H_0^{-1}\| \ge \frac{1}{t}\max_j \frac{1}{|f_j|}$: $\min_j |f_j| \sim 0$?

$$\underbrace{\begin{bmatrix} f(1) & f(p) & \dots & f(p^{t-1}) \\ f(p) & f(p^2) & \dots & f(p^t) \\ \vdots & \vdots & \ddots & \vdots \\ f(p^{t-1}) & f(p^t) & \dots & f(p^{2t-2}) \end{bmatrix}}_{H_0} = \underbrace{\begin{bmatrix} 1 & \cdots & 1 \\ b_1 & \cdots & b_t \\ \vdots & & \vdots \\ b_1^{t-1} & \cdots & b_t^{t-1} \end{bmatrix}}_{V} \underbrace{\begin{bmatrix} f_1 & & & \\ & f_2 & & \\ & & \ddots & \\ & & & f_t \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} 1 & b_1 & \cdots & b_1^{t-1} \\ 1 & b_2 & \cdots & b_2^{t-1} \\ \vdots & & & \vdots \\ 1 & b_t & \cdots & b_t^{t-1} \end{bmatrix}}_{V^{\mathrm{Tr}}}$$

☹ Difficult to find roots in $G(z)$ if $b_1, \dots, b_t$ are clustered.
(and when they're not...)

Moreover, in Ben-Or/Tiwari algorithm

Recover numerical multivariate terms?

On the unit circle: primitive roots of unity

$$f(x) = \sum_{1 \le j \le t} f_j x^{d_j}, \quad p > \deg f, \quad \omega = e^{2\pi i/p}, \beta_j = \omega^{d_j}$$

$$\underbrace{\begin{bmatrix} f(1) & f(\omega) & \dots & f(\omega^{t-1}) \\ f(\omega) & f(\omega^2) & \dots & f(\omega^t) \\ \vdots & \vdots & \ddots & \vdots \\ f(\omega^{t-1}) & f(\omega^t) & \dots & f(\omega^{2t-2}) \end{bmatrix}}_{H_0} = \underbrace{\begin{bmatrix} 1 & \cdots & 1 \\ \beta_1 & \cdots & \beta_t \\ \vdots & & \vdots \\ \beta_1^{t-1} & \cdots & \beta_t^{t-1} \end{bmatrix}}_{V} \underbrace{\begin{bmatrix} f_1 & & & \\ & f_2 & & \\ & & \ddots & \\ & & & f_t \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} 1 & \beta_1 & \cdots & \beta_1^{t-1} \\ 1 & \beta_2 & \cdots & \beta_2^{t-1} \\ \vdots & & & \vdots \\ 1 & \beta_t & \cdots & \beta_t^{t-1} \end{bmatrix}}_{V^{\mathrm{Tr}}}$$

☺ Better for conditioning:

$$\|H_0^{-1}\| \le t \cdot \frac{\|(V^{\mathrm{Tr}})^{-1}\|^2}{|f_j|} \quad \text{and} \quad \|(V^{\mathrm{Tr}})^{-1}\| \le \frac{\sqrt{2t}}{\max_k \prod_{j \ne k} |\beta_j - \beta_k|}$$

☺ Root finding is easier: perturbing k-th coefficient of f by ε changes a root β_j by $\varepsilon \cdot |\beta_j|^k / f'(\beta_j)$

☺ The exponents d_j can be recovered: powers of the primitive root of unity.

However, β_j may still be clustered on the unity circle

$\Longrightarrow V$ poorly conditioned $\Longrightarrow H_0$ poorly conditioned

Randomly choose a root of unity: $\bar{\omega} = e^{2s\pi i/p}$

random $s \in \{1, \ldots, p-1\}, \quad f(x) = \sum_{1 \le j \le t} f_j x^{d_j}, \quad \bar{\beta}_j = \bar{\omega}^{d_j}$

With high probability, $\bar{\omega}^{d_j}$ distributed relatively uniformly around the unit circle.

$$\underbrace{\begin{bmatrix} f(1) & f(\bar{\omega}) & \ldots & f(\bar{\omega}^{t-1}) \\ f(\bar{\omega}) & f(\bar{\omega}^2) & \ldots & f(\bar{\omega}^t) \\ \vdots & \vdots & \ddots & \vdots \\ f(\bar{\omega}^{t-1}) & f(\bar{\omega}^t) & \ldots & f(\bar{\omega}^{2t-2}) \end{bmatrix}}_{H_0} = \underbrace{\begin{bmatrix} 1 & \cdots & 1 \\ \bar{\beta}_1 & \cdots & \bar{\beta}_t \\ \vdots & & \vdots \\ \bar{\beta}_1^{t-1} & \cdots & \bar{\beta}_t^{t-1} \end{bmatrix}}_{V} \underbrace{\begin{bmatrix} f_1 & & & \\ & f_2 & & \\ & & \ddots & \\ & & & f_t \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} 1 & \bar{\beta}_1 & \cdots & \bar{\beta}_1^{t-1} \\ 1 & \bar{\beta}_2 & \cdots & \bar{\beta}_2^{t-1} \\ \vdots & & & \vdots \\ 1 & \bar{\beta}_t & \cdots & \bar{\beta}_t^{t-1} \end{bmatrix}}_{V^{\mathrm{Tr}}}$$

☺ With high probability: $\prod_{j \ne k} |\bar{\beta}_j - \bar{\beta}_k| \approx \frac{1}{\sqrt{t}}$ (not too small)

$\Longrightarrow V$ well-conditioned $\Longrightarrow H_0$ better-conditioned

Multivariate case:
primitive roots of unity with orders relatively prime

$\omega_k = \exp(2 s_k \pi i / p_k)$ and $f(\omega_1^i, \ldots, \omega_n^i)$

$m = p_1 \cdots p_n, \qquad p_1, \ldots, p_n$ relatively prime.

- Each term β_j is a power of $\omega_m = \exp(2\pi i/m)$

  $\Longrightarrow$ Compute $d = \operatorname{round}(\log_{\omega_m} \beta_j)$
- Recover exponents $d_{1,j}, \ldots, d_{n,j}$ by the (reverse) Chinese remainder algorithm:

$$d = d_{1,j} \cdot s_1 \cdot \left(\frac{m}{p_1}\right) + \cdots + d_{n,j} \cdot s_n \cdot \left(\frac{m}{p_n}\right)$$

☺ Numerical multivariate terms can be recovered.

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.

Generalized eigenvalue reformulation

Golub, Milanfar, and Varah 1999

For $F(x) = \sum_{1 \le j \le t} F_j e^{\delta_j x}$

$$\underbrace{\begin{bmatrix} F(1) & \dots & F(t) \\ \vdots & \ddots & \vdots \\ F(t) & \dots & F(2t-1) \end{bmatrix}}_{H_1} v = \beta \underbrace{\begin{bmatrix} F(0) & \dots & F(t-1) \\ \vdots & \ddots & \vdots \\ F(t-1) & \dots & F(2t-2) \end{bmatrix}}_{H_0} v$$

has (eigenvalue) solutions $e^{\delta_1}, \dots, e^{\delta_t}$ for β

☺ avoid construction of the generating polynomial $G(z)$

☺ avoid root finding on generating polynomial $G(z)$

☺ the generalized eigenvalue problem has numerically more stable algorithms (even when not at roots of unity).

Sparse interpolation via generalized eigenvalues

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

$$\underbrace{\begin{bmatrix} f(p^0) & f(p) & \dots & f(p^{t-1}) \\ f(p) & f(p^2) & \dots & f(p^t) \\ \vdots & \vdots & \ddots & \vdots \\ f(p^{t-1}) & f(p^t) & \dots & f(p^{2t-2}) \end{bmatrix}}_{H_0} v = \beta \underbrace{\begin{bmatrix} f(p) & f(p^{t+1}) & \dots & f(p^t) \\ f(p^2) & f(p^3) & \dots & f(p^{t+1}) \\ \vdots & \vdots & \ddots & \vdots \\ f(p^t) & f(p^{t+1}) & \dots & f(p^{2t-1}) \end{bmatrix}}_{H_1} v$$

- Solutions for $\beta : \tilde{\beta}_1, \tilde{\beta}_2, \dots, \tilde{\beta}_t$ approximate $p^{d_1}, p^{d_2}, \dots, p^{d_t}$.
- Obtain candidates for $d_1 = \text{round}(\log_p \tilde{\beta}_1), \dots, d_t = \text{round}(\log_p \tilde{\beta}_t)$
- Compute f_j by solving the associated Vandermonde system

Sparsity t in the target polynomial f ?

Binary search

guess $T \geq t$...

Cabay-Meleshko algorithm (1993)

detect first badly conditioned leading minor of H_0.

Knowledge of sparsity t may not necessary

an upper bound $T \geq t$ may still lead to a good result.

Other issues

Different bases:

can develop for Chebyshev, Pochhammer, and other bases.

Sensitivity analysis:

full sensitivity analysis to obtain stronger guarantees on output (especially degrees.)

Better results for sample points over $\mathbb{R}$:

blocking and other techniques may obtain better stability.