The Ben-Or Tiwari Sparse Interpolation Algorithm (1988)

October 14, 2021 12:13 PM

Let $f = \sum_{i=1}^{t} a_i M_i(x_1, \ldots, x_n)$ where $a_i \in \mathbb{Z}$ and $d_i = \deg(f, x_i)$.

Zippel's sparse interpolation algorithm from 1979 needs $\left(\sum (d_i + 2)\right) \cdot t$ points in $\mathbb{Z}$ to interpolate f w.h.p.

The Ben-Or & Tiwari algorithm needs $2t+2$ points w.h.p.

Both algs. need to be modified to work mod p for efficiency.

① Assume given $T \geq t$. [Not practical]

Compute $V_j = f(2^j, 3^j, 5^j, \ldots, p_n^j)$ for $0 \leq j \leq 2T-1$. [Any primes]

Let $m_i = M_i(2, 3, 5, \ldots, p_n) \in \mathbb{Z}$ be the monomial evaluations. [m_i distinct]

Suppose we can determine the m_i and t.

If $\boxed{M_i = x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}}$ then $\boxed{m_i = 2^{d_1} 3^{d_2} \cdots p_n^{d_n}}$ so $d_1, d_2, \ldots, d_n$ can be determined from m_i by dividing m_i by $2, 3, 5, \ldots, p_n$. E.g. $m_i = 300 = 2^2 \cdot 3^1 \cdot 5^2 \Rightarrow M_i(x_1, x_2, x_3) = x_1^2 \cdot x_2 \cdot x_3^2$.

How do we determine a_i?

$$M_i(2^j, 3^j, \ldots, p_n^j) = (2^j)^{d_1} (3^j)^{d_2} \cdots (p_n^j)^{d_n}$$
$$= (2^{d_1})^j (3^{d_2})^j \cdots (p_n^{d_n})^j$$
$$= m_i^j$$

$$\Rightarrow f(2^j, 3^j, \ldots, p_n^j) = \boxed{\sum_{i=1}^{T} a_i \cdot m_i^j = V_j} \text{ for } 0 \leq j \leq 2T-1.$$

Consider

$$\underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \\ m_1 & m_2 & \cdots & m_t \\ m_1^2 & m_2^2 & \cdots & m_t^2 \\ & \vdots & & \\ m_1^{t-1} & m_2^{t-1} & \cdots & m_t^{t-1} \end{bmatrix}}_{V} \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_t \end{bmatrix}}_{a} = \underbrace{\begin{bmatrix} V_0 \\ V_1 \\ \vdots \\ V_{t-1} \end{bmatrix}}_{v}$$

distinct monomial evaluations

$\prod (m_i - m_j) \Rightarrow V^{-1}$ exists.

$$\begin{bmatrix} m_1^{t-1} & m_2^{t-1} & \cdots & m_t^{t-1} \end{bmatrix} \begin{bmatrix} a_t \end{bmatrix} \quad \begin{bmatrix} v_{t-1} \end{bmatrix}$$

V^T is a Vandermonde matrix $\Longrightarrow \det(V) = \prod_{1 \le j < i \le t} (m_i - m_j) \Rightarrow V^{-1}$ exists.

$Va = v$ can be solved in $O(t^2)$ ops. in $\mathbb{Q}$.

(2) How do we determine m_i and t?

Let $\lambda(z) = \prod_{i=1}^{t} z - \underset{\in \mathbb{Z}}{m_i} = \boxed{\lambda_0 + \lambda_1 \cdot z + \cdots + \lambda_{t-1} z^{t-1} + z^t} \in \mathbb{Z}[z]$.

We will solve for λ_i then compute the roots m_i of $\lambda(z)$ by factoring $\lambda(z)$ mod $p > m_i \le p_n^d$, $d = \deg(f)$. using Cantor–Zassenhaus. in $\mathbb{Z}_p[z]$.

(3) Let $\lambda(z) = \sum_{j=0}^{t} \lambda_j z^j = \prod_{i=1}^{t} (z - m_i)$

Consider $\sum_{i=1}^{t} a_i m_i^{\ell} \underbrace{\lambda(m_i)}_{=0} = 0 = \sum_{i=1}^{t} a_i m_i^{\ell} \left(\sum_{j=0}^{t} \lambda_j m_i^j \right)$ for $\ell = 0, 1, \ldots$

$$= \sum_{i=1}^{t} \sum_{j=0}^{t} a_i m_i^{\ell+j} \cdot \lambda_j = \sum_{j=0}^{t} \lambda_j \left(\sum_{i=1}^{t} a_i m_i^{\ell+j} \right)$$

$$= \overset{j=0}{\lambda_0} \underbrace{\boxed{\sum_{i=1}^{t} a_i m_i^{\ell}}}_{v_\ell} + \overset{j=1}{\lambda_1} \underbrace{\boxed{\sum_{i=1}^{t} a_i m_i^{\ell+1}}}_{v_{\ell+1}} + \cdots + \overset{j=t}{\underset{1}{\lambda_t}} \underbrace{\boxed{\sum_{i=1}^{t} a_i m_i^{\ell+t}}}_{v_{\ell+t}} = 0.$$

$\Rightarrow \lambda_0 v_\ell + \lambda_1 v_{\ell+1} + \cdots + \lambda_{t-1} v_{\ell+t-1} = -v_{\ell+t}$ for $\ell = 0, 1, 2, \ldots$

These are linear equations in λ_i.

$$\begin{array}{c} \ell=0 \\ \ell=1 \\ \\ \\ \ell=t-1 \end{array} \underbrace{\begin{bmatrix} v_0 & v_1 & \cdots & v_{t-1} \\ v_1 & v_2 & \cdots & v_t \\ & \vdots & & \\ v_{t-1} & v_t & \cdots & v_{2t-2} \end{bmatrix}}_{H_t} \underbrace{\begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_{t-1} \end{bmatrix}}_{\lambda} = \underbrace{\begin{bmatrix} -v_t \\ -v_{t+1} \\ \vdots \\ -v_{2t-1} \end{bmatrix}}_{S}.$$

Solving $H_t \lambda = S$ can be solved using the Berlekamp–Massey alg. or Euc. Alg. in $O(t^2)$ ops.

Theorem. If $T \ge t$ then $\operatorname{rank}(H_T) = t$.

H_t is called a Hankel matrix

Problem: Let $d = \deg(f)$.

$v_j = f(\underline{2^j, 3^j, \ldots, p_n^j})$ for $0 \le j \le 2T-1$.

$v_j \approx \underline{p_n}^{d \cdot 2T}$ very big integers.

$\uparrow$ 100 $\uparrow$ 1000

Solution. Pick $p > m_i \le p_n^d$ $\quad M = x_n^d$

Do the steps mod p.

Using fast multiplication in $\mathbb{Z}_p[z]$

① Solving $Va = v$ is $O(M(t)\log t)$ ops in $\mathbb{Z}_p$.

② Factoring $\lambda(z)$ is $O([\underbrace{M(t)\log p}_{\text{POWMOD}} + \underbrace{M(t)\log t}_{\text{GCD.}}]\log t)$ ops.

③ Solving $H_t \lambda = s$ is $O(M(t)\log t)$ using the fast E.A.