Multivariate Multipoint Evaluation (MME)

Sarwagya Prasad, Rahul Bhardwaj

August 7, 2024

Content

Introduction

Univariate Multipoint Evaluation over Finite Fields

Preliminaries 1

Definition

Define the precision function for integers and polynomials as follows :

$$\mathrm{prec}(U) = \begin{cases} \deg U + 1 & \text{if } U \text{ is a polynomial,} \\ \log_2 U & \text{if } U \text{ is an integer.} \end{cases}$$

PolyMult

Multiplication of two polynomials of degree n and m takes :

$$\frac{9}{2} N \log N + 5N + \mathrm{Idt}$$

time, where $N = n + m$.

Preliminaries 2

Divide and Rule

If the timing function of an algorithm satisfies the recurrence :

$$T(N) = 2T(N/2) + f(N)$$

, where $f(N) = \mathcal{O}(N \log^a(N))$, then $T(N) = \mathcal{O}(N \log^{a+1}(N))$.

Evaluation is Division

1. Let $p(X) \in \mathbb{F}[X]$ be a polynomial which we want to evaluate at $x = \alpha \in \mathbb{F}$.
2. By division algorithm, there exist $q(X), r(X) \in \mathbb{F}[X]$ such that $\deg r < \deg(x - \alpha)$:
$$p(X) = q(X)(x - \alpha) + r(x).$$
3. Hence $p(\alpha) = r(\alpha)$, but since $\deg r = 0$, r is a constant polynomial.
4. Therefore, evaluating a polynomial at a point α becomes a problem of how quickly you can find the remainder of the corresponding division of the polynomial by $x - \alpha$.

Evaluation is Division

If we want to evaluate a polynomial $p(x)$ and $x_1, \ldots, x_N$, then we try to write $f(x) = q(x)\prod_{1 \leq i \leq N}(x - x_i) + r(x)$, where $< N$. Hence, $f(x_i) = r(x_i)$ for $1 \leq i \leq N$. Therefore, we've reduced our problem to a simpler problem.It suffices to consider the problem of evaluating polynomials of degree $N - 1$ on N points. Let $M_1(x) = (x - x_1)\cdots(x - x_{N/2})$ and $M_2(x) = (x - x_{N/2+1})\cdots(x - x_N)$. We divide $p(x)$ by $M_1(x)$ to get $R_1(x)$ and by $M_2(x)$ to get $R_2(x)$. The problem now reduces to evaluating two polynomials of $N/2 - 1$ degree at $N/2$ points each.

Algorithm

Let D be a Euclidean domain, we are given a set of N moduli $\{m_i\} \in D$ and an element $U \in D$ for which we wish to compute the set of residues $u_i \in D$ such that :

$$u_i \equiv U \pmod{m_i}, \quad 1 \leq i \leq N.$$

Theorem

Given N moduli $m_i \in D$ and $U \in D$ where $\text{prec}(U) = N$, if multiplication and division of N precision elements can be performed in $\mathcal{O}(N \log^a(N))$ operations, then the N residues $\{u_i\}$ of U, with respect to $\{m_i\}$ can be computed in $\mathcal{O}(N \log^{a+1}(N))$ steps, where $a \geq 0$.

Algorithm

$\boxed{\textsc{Modular Form}(\mathrm{U,j,k})}$

Input :

- the requisite moduli M_{jk},
- the element U where $\mathrm{prec}(U) \leq k - j + 1$.

Output : the residues $u_i \equiv U \bmod m_i, \quad j \leq i \leq k$.

Step

1. If $j = k$, then output U and go to step 4.
2. Let $e := \lfloor (j + k - 1)/2 \rfloor$ and $f := e + 1$. Set $R_1 := U \,\mathrm{rem}\, M_{je}$ and $R_2 := U \,\mathrm{rem}\, M_{fk}$.
3. Call $\textsc{Modular Forms}(R_1, j, e)$ and $\textsc{Modular Forms}(R_2, f, k)$.
4. Return.

Approximate Univariate Multipoint Evaluation over Complex

Statement

Problem Statement

Given a univariate polynomial $f \in \mathbb{C}[x]$ of degree d with $\|f\| \leq 2^{\tau}$, and d points $x_1, x_2 \cdots x_d$ with absolute value less than 1, return the approximate evaluation of f on these points, $y_1, y_2 \cdots y_k$ such that $|f(x_i) - y_i| \leq \|f\| 2^{-m}$

Nearly Linear Time

Theorem

[Mor21]: The above problem can be solved in $\tilde{O}(d(\tau + m))$ bit operations.

[Mor21]: Guillaume Moroz. New data structure for univariate polynomial approximation and appications in root isolation, numerical multipoint evaluation and other problems

Previous State of the Art

Theorem

[KS16]: Let f be a polynomial of degree d, with $||f||_1 \leq 2^{\tau}$, and let $x_1, x_2 \cdots x_d \in \mathbb{C}$ be complex points with absolute values bounded by 1. Then, computing y_k such that $|f(x_k) - y_k| \leq 2^{-m}$ is possible in $\tilde{O}(d(m+\tau+d))$ bit operations.

[KS16]: Alexander Kobel and Michael Sagraloff. Fast approximate polynomial multipoint evaluation and many applications

Unit Disk Covering

- If $m < d$, the previous algorithm is optimal.
- If we had a m degree approximation of f, we could use the previous algorithm to get the required nearly linear time algorithm.
- However, we cannot hope a single m degree approximation to stay close to f. Thus we can make a **partition**, or more generally, a covering, of the unit disk with many small parts, and have approximations g for each small part such that g stays close to f in that region.
- Need to limit the number of parts, e.g., have $O(d/m)$ parts.

N-Hyperbolic Covering

Definition

[Mor21]: Given a positive integer N, an N-hyperbolic covering of the unit disk is the set of disks of centres $\gamma_n e^{2\pi i \frac{k}{K_n}}$ and radii ρ_n, $0 \leq n < N, 0 \leq k < K_n$ where:

$$r_n = \begin{cases} 1 - \frac{1}{2^n} & , 0 \leq n < N \\ 1 & , n = N \end{cases}$$

$$\gamma_n = \frac{1}{2}(r_n + r_{n+1})$$

$$\rho_n = \frac{3}{4}(r_{n+1} - r_n)$$

$$K_n = \begin{cases} 4 & , n = 0 \\ \lceil \frac{3\pi}{\sqrt{5}} \frac{r_{n+1}}{\rho_n} \rceil & , otherwise \end{cases}$$

Illustration of Hyperbolic Covering

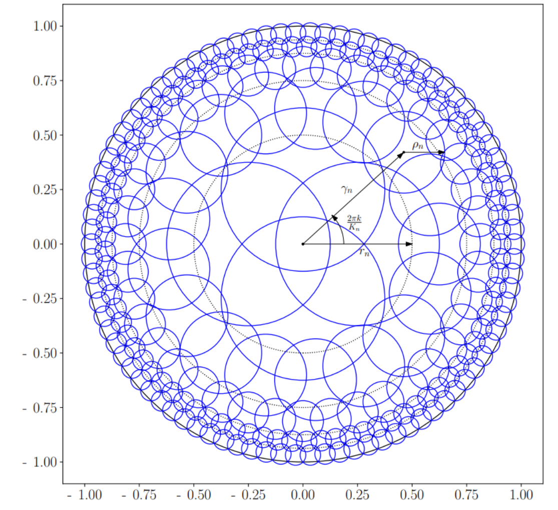

Figure 1: 5-hyperbolic covering

[Mor21]

m-hyperbolic Approximation

Given a polynomial of degree d with $\|f\| \leq 2^{\tau}$, and an integer $m > 1$, an m-hyperbolic approximation $H_{d,m}$ of f is a finite set of pairs (g, a) where g is an m degree polynomial, and a is an affine transformation such that:

- The set of disks $a\,(D(0,1))$ is the N-hyperbolic covering, with $N = \lceil \log_2 \left(\frac{3ed}{m} \rceil \right)$, i.e., $a(X) = (\gamma_n + \rho_n X)\, e^{2\pi i \frac{k}{K_n}}$
- $\|f \circ a - g\| \leq 3\|f\|2^{-m}$

[Mor21]

Number of disks

Lemma

Given two integers d and $m > 1$, let $N = \lceil \log_2 \left(\frac{3ed}{m}\right\rceil)$. Then, the number of disks in the N-hyperbolic covering is in $O(d/m)$ and the union of the disks contains the unit disk.

Proof

Total number of disks $t = \sum_0^{N-1} K_n$
We have $K_n \leq 2^{n+4}$, $\Rightarrow t \leq 2^{N=4} \leq 16 \cdot 3e\frac{d}{m}$
Therefore, $t = O(d/m)$
CLAIM: For any ring $R_n = D(0, r_{n+1}) \setminus D(0, r_n)$, the union of disks centered at $\gamma e^{2\pi i \frac{k}{K_n}}$ with radius ρ_n contains R_n.

[Mor21]

Approximation Algorithm

Theorem

Given a polynomial f of degree d, and an integer $m > 1$, the m-hyperbolic approximation of f can be computed in $\tilde{O}(d(m+\tau))$

Approximation Algorithm

Algorithm 1: Hyperbolic approximation data structure

Input: A polynomial $f(X) = \sum_{k=0}^{d} f_k X^k$ of degree d with $\|f\|_1 \leq 2^\tau$, $\tau \geq 1$, and an integer $m \geq 1$

Output: An m-hyperbolic approximation of f (see Definition 2)

$\widetilde{m} \leftarrow \min(m-1, d)$

$N \leftarrow \lceil \log_2(3ed/\widetilde{m}) \rceil$

for n *from* 0 *to* $N-1$ **do**

    # Compute $(g_{n,k}, a_{n,k})$ *for the disks covering* $D(0, r_{n+1}) \setminus D(0, r_n)$

    # The precision of the arithmetic operations is in $\Theta(m + \tau + \log d)$

    # A. Compute r_n, γ_n, ρ_n *and* K_n *for the* $a_{n,k}(X) = (\gamma_n + \rho_n X) e^{i2\pi \frac{k}{K_n}}$

    $r_n \leftarrow 1 - 1/2^n$

    $r_{n+1} \leftarrow 1 - 1/2^{n+1}$ if $n \leq N-2$ else 1

    $\gamma_n \leftarrow (r_n + r_{n+1})/2$

    $\rho_n \leftarrow \frac{3}{4}(r_{n+1} - r_n)$

    $K_n \leftarrow \left\lceil \frac{3\pi}{\sqrt{5}} \frac{r_{n+1}}{\rho_n} \right\rceil$

    # B. Compute $g_{n,k}(X) \approx f\left((\gamma_n + \rho_n X) e^{i2\pi \frac{k}{K_n}}\right) \mod X^m$

    # B.1. Truncate f *at* d_n *such that* $(\gamma_n + \rho_n)^{d_n+1} \leq 1/2^{m+1}$

    $d_n \leftarrow \min\left(d, \lceil \frac{8}{3} \log(2)(m+1)2^n \rceil - 1\right)$ if $n < N-1$ else d

    $p \leftarrow f_0 + \cdots + f_{d_n} X^{d_n}$

    # B.2. Gather the coefficients in Y *of* $p(YZ) \mod Z^{K_n} - 1$,

    # where Y *and* Z *are symbolic variables*

    for k *from* 0 *to* $K_n - 1$ **do**

        $p_k(Y^{K_n})Y^k \leftarrow$ coefficients of Z^k of $p(YZ) \mod Z^{K_n} - 1$

[Mor21]

Approximation Algorithm

$q_0(X) \leftarrow 1$

for k *from* 1 *to* K_n **do**

  $q_k(X) \leftarrow q_{k-1}(X) \cdot (\gamma_n + \rho_n X) \mod X^{\overline{m}}$

`# B.4. Compute` $r_k(X) = p_k\left((\gamma_n + \rho_n X)^{K_n}\right) \cdot (\gamma_n + \rho_n X)^k \mod X^{\overline{m}}$

for k *from* 0 *to* $K_n - 1$ **do**

  $r_{k,0} + \cdots + r_{k,\overline{m}-1} X^{\overline{m}-1} \leftarrow p_k(q_{K_n}(X)) \cdot q_k(X) \mod X^{\overline{m}}$

`# B.5. Compute` $g_{n,k}(X) = r_0(X) + \cdots + r_{K_n-1}(X) e^{i2\pi \frac{k}{K_n}(K_n-1)}$

for ℓ *from* 0 *to* $\overline{m} - 1$ **do**

  $s_\ell(Z) \leftarrow r_{0,\ell} + \cdots + r_{K_n-1,\ell} Z^{K_n-1}$

  $g_{n,0,\ell}, \ldots, g_{n,K_n-1,\ell} \leftarrow s_\ell(e^{i2\pi\frac{0}{K_n}}), \ldots, s_\ell(e^{i2\pi\frac{K_n-1}{K_n}})$

`# B.6. Append the pair to the result list`

for k *from* 0 *to* $K_n - 1$ **do**

  $g_{n,k}(X) \leftarrow g_{n,k,0} + \cdots + g_{n,k,\overline{m}-1} X^{\overline{m}-1}$

  $a_{n,k}(X) \leftarrow (\gamma_n + \rho_n X) e^{i2\pi\frac{k}{K_n}}$

  Append the pair $(g_{n,k}, a_{n,k})$ to the list L

return L

[Mor21]

Overall Algorithm

Algorithm 1: APPROX-MULTIPOINT-EVAL Algorithm

Data: polynomial f of degree d, d numbers $x_i \in D(0,1)$, precision m

Result: List of evaluations y_i such that $|y_i - f(x_i)| \leq \|f\| 2^{-m}$

```
L ← {}
Q ← data structure constructed from x_i, for fast disk range search
G ← H_{d,m+2}(f)
for (g_k, a_k) in G do
    v_1, ··· v_{n_k} ← query Q for range a_k
    y_1, ··· y_{n_k} ← g_k(a_k^{-1}(v_1)) ··· g_k(a_k^{-1}(v_{n_k}))
    Append y_1, ··· y_k to L
end
return L;
```

Multivariate Multipoint Evaluation over Finite Fields

Preliminaries - CRT

Lemma

Fast CRT Modulation Computation [GG13]: There is an algorithm that when given as input coprime positive integeres $p_1, \cdots p_r$ and a positive integer $N < \Pi p_i < 2^c$ computes the remainders $a_i \equiv N \bmod p_i$ in $\tilde{O}(c)$ time

Lemma

Fast CRT Reconstruction [GG13]: There is an algorithm that when given input coprime positive integers $p_1, \cdots p_r$ and $a_1 \cdots a_r$ such that $0 \leq a_i < p_i$ outputs the unique integer $0 \leq N < \Pi p_i < 2^c$ such that $N \equiv a_i \bmod p_i$ in $\tilde{O}(c)$ time.

[GG13]: Joachim Von Zur Gathen and Jurgen Gerhard: Modern Computer Algebra

Preliminaries - Kronecker Map

Definition

Kronecker Map: The c-variate Kronecker Map for base-d denoted by $\Phi_{d,m;c}$ maps cm-variate polynomials into a c-variate polynomials via:

$$\Phi_{d,m;c}(f(x_{11},\cdots x_{1m},\cdots x_{cm})) = f(1, y_1^d, y_1^{d^2}\cdots y_1^{d^{m-1}},\cdots y_c^{d^{m-1}})$$

Frame Title

Theorem

Given m-variate polynomial $f \in \mathbb{F}_p[x_1, \cdots x_m]$ with degree at most d-1 in each variable and $\alpha_1 \cdots \alpha_{N-1}$, then there exists a deterministic algorithm that outputs $f(i)$ in time:

$$O\left(m(d^m + p^m + N)poly(\log p)\right)$$

Proof

1. Compute the reduction $\bar{f}$ of f modulo $x_j^p - x$
2. Use an FFT to compute $\bar{f}(\alpha)$ for all $\alpha \in \mathbb{F}_p^m$
3. Look up and return $f(\alpha_i)$

MME Algorithm over Finite Fields

Algorithm 2: MME-FINITE-FIELD

Data: $f(x_1, \ldots, x_n) \in \mathbb{F}_p[x_1, \ldots, x_n]$ and $\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(N)} \in \mathbb{F}_p^N$

Result: $b_i = f(\mathbf{a}^{(i)})$ for $i \in [N]$.

```
Adjust d and m such that log log d ≤ m ≤ d^{o(1)} Let L̃ = (dm + 1)logp + mlogd. Compute first L̃ prime
  numbers {p_1, ··· p_L̃}
Let L ≤ s be the smallest integer such that p_1 ··· p_L =: M > d^m · p^{dm+1}
for e ∈ {0, ..., d − 1}^m do
    Compute f_e^{(l)} = f_e mod p_l for l ∈ L.
end
for i ∈ [N], k ∈ [m] do
    Compute a_{i,k,l} = a_k^{(l)} mod p_l for l ∈ L.
end
for l ∈ [L] do
    Let f^{(l)}(x_1, ..., x_m) = Σ_e f_e^{(l)} x^e ∈ F_{p_l}[x].
    Let a^{(i,l)} = (a_{i,1,l}, ..., a_{i,m,l}) ∈ F_{p_l}^m for each i ∈ [N].
    Compute f_{(l)}(α) for all α ∈ F_p^m
    Look up and store f^{(l)}(a^{(i,l)})
end
for i ∈ [N] do
    Compute the unique b_i ∈ [−M/2, M/2] such that b_i = b_{i,l} mod p_l for all l ∈ [L].
end
return {b_i : i ∈ [N]};
```

Multivarate Multipoint Evaluation over Integers

Exact-MME over $\mathbb{Z}$

Problem Statement

Input : An integer $s > 0$, a polynomial $f(x_1, \ldots, x_n) \in \mathbb{Z}[x_1, \ldots, x_m]$ of individual degree less than d, given as a list of d^m integer coefficients, a set of points $\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(N)} \in \mathbb{Z}^m$ with each coordinate of magnitude at most 2^s, with the guarantee that all coefficients of f, coordinates of $\mathbf{a}^{(i)}$'s, and evaluations $f(\mathbf{a}^{(i)})$ are bounded in magnitude by 2^s.

Output : Integers $b_1, \ldots, b_N$ that are the evaluations, i.e. $b_i = f(\mathbf{a}^{(i)})$ for $i \in [M]$.

Exact-MME over $\mathbb{Z}$

Theorem

There is a deterministic algorithm that on input as mentioned above returns the required output as mentioned above and runs in deterministic time $((d^m + Nm) \cdot s)^{1+o(1)}$ for all $m \in \mathbb{N}$ and sufficiently large $d \in \mathbb{N}$.

Algorithm

Algorithm 3: EXACT-MME-INTEGERS

Data: $f(x_1, \dots, x_n) \in \mathbb{Z}[x_1, \dots, x_n]$ and $\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(N)} \in \mathbb{Z}^N$, and an integer $s > 0$ such that $|\mathbf{a}^{(i)}| < 2^s$ and $|f(\mathbf{a}^{(i)})| < 2^s$.

Result: $b_i = f(\mathbf{a}^{(i)})$ for $i \in [N]$.

Compute the first s prime numbers $\{p_1, \dots, p_s\}$.
Let $L \leq s$ be the smallest integer such that $p_1 \cdots p_L =: M > 2^{s+1}$.
for $e \in \{0, \dots, d-1\}^m$ **do**
  Compute $f_e^{(l)} = f_e \bmod p_l$ for $l \in L$.
end
for $i \in [N]$, $k \in [M]$ **do**
  Compute $a_{j,k,l} = \mathbf{a}_k^{(l)} \bmod p_l$ for $l \in L$.
end
for $l \in [L]$ **do**
  Let $f^{(l)}(x_1, \dots, x_m) = \sum_{\mathbf{e}} f_{\mathbf{e}}^{(l)} \mathbf{x}^{\mathbf{e}} \in \mathbb{F}_{p_l}[\mathbf{x}]$.
  Let $\mathbf{a}^{(i,l)} = (a_{i,1,l}, \dots, a_{i,m,l}) \in \mathbb{F}_{p_l}^m$ for each $i \in [N]$.
  Compute $b_{i,l} = f^{(l)}(\mathbf{a}^{(i,l)})$ for all $i \in [N]$ using Finite MME algorithm.
end
for $i \in [N]$ **do**
  Compute the unique $b_i \in [-M/2, M/2]$ such that $b_i = b_{i,l} \bmod p_l$ for all $l \in [L]$.
end
return $\{b_i : i \in [N]\}$;