Fast Polynomial Factorization And Modular Composition

Ashish Dwivedi

IIT Kanpur

April 15, 2017

Table of Contents

Introduction

- This is work of Kedlaya and Umans[2008].
- A randomized algorithm for factoring degree n univariate polynomial over $\mathbb{F}_q$ taking $O(n^{1.5+o(1)} \log^{1+o(1)} q + n^{1+o(1)} \log^{2+o(1)} q)$ bit operations.
- For $\log q < n$ this is asymptotically fastest algorithm and for $\log q \geq n$ it is same as best previous algorithms [von zur Gathen, Shoup [GS92] and Kaltofen, Shoup [KS98]].

Idea

- Asmptotic bottleneck in GS92 and KS98 is "Modular Composition" (MOC) of univariate polynomials of degree n.
- This work improves MOC and hence the above factoring algorithms.
- Complexities of previous works for MOC were dependent over the exponent of matrix multiplication.
- This work gives a different approach to solve MOC by reducing it to "Multivariate Multipoint Evaluation" (MME) problem.
- It solves MME by lifting it to $\mathbb{Z}$, applying small number of multimodular reduction and then completing with a small number of multidimensional FFTs.

Problem Statements

We formally define the problems MOC and MME.

Modular Composition

Given $f(X_0, \ldots, X_{m-1})$ in $R[X_0, \ldots, X_{m-1}]$ with individual degrees at most $d-1$, and polynomials $g_0(X), \ldots, g_{m-1}(X)$ and $h(X)$, all in $R[X]$ with degree at most $N-1$, and with the leading coefficient of h invertible in R, output $f(g_0(X), \ldots, g_{m-1}(X)) \bmod h(X)$.

This is a slightly generalized version of simple modular composition.

Multivariate Multipoint Evaluation

Given $f(X_0, \ldots, X_{m-1})$ in $R[X_0, \ldots, X_{m-1}]$ with individual degrees at most $d-1$, and evaluation points $\alpha_0, ..., \alpha_{N-1}$ in R^m, output $f(\alpha_i)$ for $i = 0, 1, 2, ..., N-1$.

Some Facts

Inverse Kronecker substitution

The map $\psi_{h,l}$ from $R[X_0, X_1, ..., X_{m-1}]$ to $R[Y_{0,0}, ..., Y_{m-1,l-1}]$ is defined as follows. Given X^a, write a in base h: $a = \sum_{j \geq 0} a_j h^j$ and define the monomial $M_a(Y_0, ..., Y_{l-1}) := Y_0^{a_0} Y_1^{a_1} \dots Y_{l-1}^{a_{l-1}}$.

- The map $\psi_{h,l}$ sends X_i^a to $M_a(Y_{i,0}, ..., Y_{i,l-1})$ and extends multilinearly to $R[X_0, X_1, ..., X_{m-1}]$.
- Note that this map is injective for the polynomials having individual degrees at most $h^l - 1$.

Number theory fact

For all integers $N \geq 2$, the product of the primes less than or equal to $16 \log N$ is greater than N.

Reduction

We first reduce MOC to MME.

Theorem 1

Given $f(X_0, ..., X_{m-1})$ in $R[X_0, ..., X_{m-1}]$ with individual degrees at most $d-1$, and polynomials $g_0(X), ..., g_{m-1}(X)$ and $h(X)$, all in $R[X]$ with degree at most $N-1$, and with the leading coefficient of h invertible in R, there is, for every $2 \leq d_0 < d$, an algorithm that outputs

$$f(g_0(X), ..., g_{m-1}(X)) \bmod h(X)$$

in $O(((d^m + mN)d_0).poly \log(d^m + mN))$ ring operations and one invocation of MME with parameters $d_0, m' = lm, N' = Nmld_0$, where $l = \lceil \log_{d_0} d \rceil$, provided that the algorithm is supplied with N' distinct elements of R whose differences are units in R.

Reduction from MOC to MME Cont..

Algorithm

- Compute $f' = \psi_{d_0,l}(f)$.
- Compute $g_{i,j}(X) := g_i(X)^{d_0^j} \bmod h(X)$ for all i and $j = 0, \dots, l-1$.
- Select N' distinct element of $R, \beta_0, \dots, \beta_{N'-1}$, whose differences are units in R. Compute $\alpha_{i,j,k} := g_{i,j}(\beta_k)$ for all i, j, k using fast (univariate) multipoint evaluation.
- Compute $f'(\alpha_{0,0,k}, ..., \alpha_{m-1,l-1,k})$ for $k = 0, \dots, N'-1$.
- Interpolate to recover $f'(g_{0,0}(X), ..., g_{m-1,l-1}(X))$ (which is a univariate polynomial of degree less than N') from these evaluations.
- Output the result modulo $h(X)$.

We can see that $f'(g_{0,0}(X), ..., g_{m-1,l-1}(X)) \equiv f(g_0(X), ..., g_{m-1}(X)) \bmod h(X)$.

Fast Multivariate Multipoint Evaluation

Over Prime fields

Given $f(X_0, \ldots, X_{m-1})$ in $\mathbb{F}_p[X_0, \ldots, X_{m-1}]$ with individual degrees at most $d-1$, and evaluation points $\alpha_0, ..., \alpha_{N-1}$ in $\mathbb{F}_p^m$,there is deterministic algorithm that outputs $f(\alpha_i)$ for $i = 0, 1, 2, ..., N-1$ in

$$O(m(d^m + p^m + N)poly(logp))$$

bit operations.

Algorithm

- Compute reduction $\bar{f}$ of f modulo $X_j^p - X_j$ for all $j \in [m-1]$.
- Use FFT to compute $\bar{f}(\alpha) = f(\alpha)\ \forall \alpha \in \mathbb{F}_p^m$.
- Look up and return $f(\alpha_i)$'s.

Fast Multivariate Multipoint Evaluation Cont..

Over Rings $\mathbb{Z}/r\mathbb{Z}$

Here we will apply t rounds of multimodular reduction. So algorithm for this takes additional parameter t (which is actually a small constant).

Algorithm Multimodular($f, \alpha_0, \ldots, \alpha_{N-1}$,r,t)

- Consider $\bar{f}$, the version of f over $\mathbb{Z}$ and also $\bar{\alpha}_i$ the version of α over $\mathbb{Z}^m$.
- Compute primes $p_1, \ldots, p_k$ less than or equal to $l = 16\log(d^m(r-1)^{md}$.
- Compute reduction $f_h = \bar{f}$ mod p_h and $\alpha_{h,i} = \bar{\alpha}_i$ mod p_h.
- If $t = 1$, for $h = 1, ..., k$ apply theorem for prime fields to compute $f_h(\alpha_{h,i})$ for $i = 0, ..., N-1$; Otherwise run this algorithm again with updated parameters p_h and $t-1$ and compute $f_h(\alpha_{h,i})$ for $i = 0, ..., N-1$.
- Apply chinese remaindering to compute $\bar{f}$ and reduce it modulo r.

Corollary 1

For every constant $\delta > 0$ there is an algorithm for MME over $\mathbb{Z}/r\mathbb{Z}$ with parameters d, m, N, and with running time $(d^m + N)^{1+\delta} \log^{1+o(1)} r$, for all d, m, N with d sufficiently large and $m \leq d^{o(1)}$.

Fast Multivariate Multipoint Evaluation Cont..

Over Extension Rings $(\mathbb{Z}/r\mathbb{Z})[Z]/(E(Z))$

Here E is a monic poly of degree e, so coefficients in this ring are poly of degree at most $e-1$ and have coefficient at most $r-1$.

Algorithm MultimodularExtension($f, \alpha_0, \ldots, \alpha_{N-1}$,t)

Let $M = d^m(e(r-1))^{(d-1)m+1}$ and $r' = M^{(e-1)dm+1}$.

- Consider $\tilde{f}$, the version of f over $\mathbb{Z}[Z]$ and also $\tilde{\alpha}_i$ the version of α_i over $\mathbb{Z}[Z]^m$.
- Compute the reduction $\bar{f}$ of $\tilde{f}$ modulo r' and $Z-M$ and reduction $\bar{\alpha}_i$ of $\tilde{\alpha}_i$ modulo r' and $Z-M$. Reduction modulo r' don't do anything computationally.
- Call Multimodular($\bar{f}, \bar{\alpha_0}, ..., \bar{\alpha}_{N-1}, r', t$) to compute $\beta_i = \bar{f}(\bar{\alpha_i})$.
- Compute unique poly $Q_i(Z) \in \mathbb{Z}[Z]$ of degree atmost $(e-1)dm$ with coefficients in $[M-1]$ for which $Q_i(M)$ has remainder β_i mod r'. Reduce it modulo r and $E(Z)$.

Fast Multivariate Multipoint Evaluation Cont..

Corollary 2

For every constant $\delta > 0$ there is an algorithm for MME over $(\mathbb{Z}/r\mathbb{Z})[Z]/(E(Z))$ of cardinality q with parameters d, m, N, and with running time $(d^m + N)^{1+\delta} \log^{1+o(1)} q$, for all d, m, N with d sufficiently large and $m \leq d^{o(1)}$.

Combine

Theorem 2

Let R be a finite ring of cardinality q given as $(\mathbb{Z}/r\mathbb{Z})[Z]/(E(Z))$ for some monic polynomial $E(Z)$. For every constant $\delta > 0$, if we have access to Nd^{δ} distinct elements of R whose differences are units in R, there is an algorithm for MOC over R with parameters d, m, N, and with running time $(d^m + N)^{1+\delta} \log^{1+o(1)} q$, for all d, m, N with d, N sufficiently large, provided $m \leq d^{o(1)}$.

Corollary 3

For every $\delta > 0$, there is an algorithm for MOC over $\mathbb{F}_q$ with parameters $d, m = 1, N = d$ running in $d^{1+\delta} \log^{1+o(1)} q$ bit operations, for sufficiently large d.

Application to Factoring over $\mathbb{F}_q$

- KS98 gives a polynomial factoring algorithm requiring $O(n^{0.5+o(1)}C(n,q) + n^{1+o(1)}\log^{2+o(1)} q)$ bit operations, where $C(n,q)$ is bit operations required for MOC of degree n polynomials over $\mathbb{F}_q$.
- Using the algorithm for MOC (Corollary 3), we get an algorithm for polynomial factorization which requires $O(n^{1.5+o(1)}\log^{1+o(1)} q + n^{1+o(1)}\log^{2+o(1)} q)$ bit operations.
- This is faster than previous algorithms GS92 and KS98 which required $(n^{2+o(1)}\log^{1+o(1)} q + n^{1+o(1)}\log^{2+o(1)} q)$ and $n^{1.815}\log^{2+o(1)} q)$ bit operations respectively, when $\log q < n$.

Thank You !