基于 Kronecker 型替换的黑盒多项式稀疏插值

献给张景中、杨路教授 85 华诞

黄巧龙¹, 高小山^{2,3*}

1. David R. Cheriton School of Computer Science, University of Waterloo, Waterloo N2L 3G1, Canada;

2. 中国科学院数学与系统科学研究院, 北京 100190;

3. 中国科学院大学数学科学学院, 北京 100049

E-mail: q94huang@uwaterloo.ca, xgao@mmrc.iss.ac.cn

收稿日期: 2019-01-10; 接受日期: 2019-04-18; 网络出版日期: 2019-10-23; * 通信作者

国家自然科学基金 (批准号: 11688101) 资助项目

摘要 本文基于新的 Kronecker 型替换, 给出两个由黑盒表示的稀疏多项式的新确定性插值算法. 令 $f \in \mathcal{R}[x_1, \dots, x_n]$ 是一个稀疏黑盒多项式, 其次数上界为 D . 当 $\mathcal{R}$ 是 $\mathbb{C}$ 或者是有限域时, 相对于已有算法, 新算法具有更好的计算复杂度或者关于 D 的复杂度更低. 特别地, 对于一般黑盒模型, D 是复杂度中的主要因素, 而在所有的确定性算法中, 本文的第二个算法的复杂度关于 D 是最低的.

关键词 稀疏插值 Kronecker 替换 黑盒模型

MSC (2010) 主题分类 41A05, 68W05, 68W30

1 引言

稀疏多项式插值是一个基本计算问题. 稀疏多项式插值算法主要分为两类: 直接插值算法^[1-5], 即直接计算多项式的各单项式的次数和系数; 约化型方法^[6-14], 即将多元多项式插值约化为单元多项式插值. 本文提出新的 Kronecker 型替换, 对于由一般标准黑盒给定的多元多项式, 提出两种新的确定性约化型插值算法.

1.1 主要结果

令 $f \in \mathcal{R}[x_1, \dots, x_n]$ 是由黑盒程序 $\mathcal{B}_f$ 所给的稀疏多项式, 其中 $\mathcal{R}$ 为可计算整环. 假设 f 的次数上界是 D , 项数上界是 T . 所谓稀疏插值就是由 $\mathcal{B}_f$ 、 T 和 D 计算 f 的显式表达式.

我们的算法是基于以下的 Kronecker 型替换:

$$f_{(d,p)}^{\text{mod}} = f(x, x^d, \dots, x^{d^{n-1}}) \bmod (x^p - 1), \quad (1.1)$$

英文引用格式: Huang Q L, Gao X S. Deterministic sparse interpolation of black-box multivariate polynomials using Kronecker type substitutions (in Chinese). Sci Sin Math, 2021, 51: 153–166, doi: 10.1360/N012019-00018

$$f_{(d,p)} = f(x, x^{\text{mod}(d,p)}, \dots, x^{\text{mod}(d^{n-1},p)}), \quad (1.2)$$

$$f_{(d,p,k)} = f(x, \dots, x^{\text{mod}(d^{k-1},p)+p}, \dots, x^{\text{mod}(d^{n-1},p)}), \quad (1.3)$$

其中 x 是一个新变元, $d, p \in \mathbb{N}_{>0}$.

第一个算法的想法是, 首先对 $d = 1, \dots, 4(n-1)(T-1) + 1$ 插值单变元多项式 $f_{(d,p)}$, 再由这些 $f_{(d,p)}$ 恢复 f , 其中 p 是一个固定的素数. 我们将这里所用的替换 $x_i = x^{\text{mod}(d^{i-1},p)}$ ($d = 1, \dots, 4(n-1)(T-1) + 1$) 称为基于基变换的 Kronecker 替换.

第二个算法的想法是, 首先对 $i = 1, \dots, N$ 插值单变元多项式 $f_{(D,p_i)}$, 然后再由这些 $f_{(D,p_i)}$ 恢复 f , 其中 N 是一个大小为 $O(nT \log D)$ 的整数, p_i 是不同的素数. 我们将这里所用的替换 $x_i = x^{\text{mod}(D^{i-1},p_k)}$ ($k = 1, \dots, N$) 称为基于模变换的 Kronecker 替换.

表 1 给出当 $\mathcal{R} = \mathbb{C}$ 时, 新算法与其他确定型算法的比较. 表 1 和 2 中的复杂度都是基于 $O\sim$ 的, 即我们忽略了一些对数因子. 为了简化结果, 假设系数的大小是 $O(1)$. 一个次数为 D 、项数为 T 、最大系数为 $O(1)$ 的 n 元多项式的大小是 $O(nT \log D)$. 由于所有插值算法关于 n 和 T 都是多项式的, 因此, 插值算法计算复杂度的主要因素是 D . 从表 1 中可以看出, 我们给出的第二个算法的复杂度关于 D 最低. 另一方面, Ben-Or & Tiwari 算法具有最低的查询次数. 这里不比较概率算法, 也不与比黑盒模型特殊的计算模型的插值算法比较, 如 straight-line programs (SLP) 模型 [7–9, 12, 15–17].

表 2 给出当 $\mathcal{R} = \mathbb{F}_q$ 时, 新算法与其他确定型算法的比较. “ $\mathbb{F}_q$ 的规模”是指该算法可以对满足该条件的有限域进行插值. 如果算法可在 $\mathbb{F}_q$ 的扩域上取值, 那么我们的算法可适用于任何有限域. 从表 2 中可看出, 我们的第一个算法比文献 [13] 中的算法复杂度低. 文献 [18] 假设 f 满足 $\deg(f, x_i) < q-1$. 在该假设下, $D \leq n(q-1)$, 我们的第二个算法的位复杂度是 $O\sim(n^3 T^2 q)$. 可以看出, 新算法关于复杂度的主要因素 q 是线性的, 而在文献 [18] 中算法的复杂度关于 $q^{2.5}$ 是线性的. 我们的第二个算法在有限域情形关于 D 同样具有最好的复杂度. 同样, 这里没有与概率算法以及比黑盒模型特殊的计算模型进行比较.

为了阅读的方便, 我们将文中出现的一些符号列在表 3 中.

表 1 $\mathbb{C}$ 上确定性插值算法的比较

	B_f 的查询次数	算术步数	数据长度	位复杂度
文献 [1, 3]	T	$nT^2 D$	TD	$nT^3 D^2$
文献 [14]	$nT^2 D$	$nT^3 D$	TD	$nT^4 D^2$
文献 [13]	nT^3	$nT^3 \log^2 D$	$TD \max(nT^2, D)$	$nT^4 D \max(nT^2, D)$
本文 (推论 3.2)	nT^2	$nT^2 \log^3 D$	$TD \max(nT, D)$	$nT^3 D \max(nT, D)$
本文 (推论 4.2)	$nT^2 \log D$	$nT^2 \log^3 D$	$nT^2 D$	$n^2 T^4 D$

表 2 $\mathbb{F}_q$ 上确定性插值算法的比较

	B_f 的查询次数	位复杂度	$\mathbb{F}_q$ 的规模
文献 [18]		$n^2 T^6 \log^2(q) + q^{2.5} \log^2 q$	$q \geq O\sim(n^2 T^2)$
文献 [13]	nT^3	$nT^2 D \max(nT^2, D) \log q$	$q \geq O\sim(D \max(nT^2, D))$
本文 (推论 3.3)	nT^2	$nTD \max(nT, D) \log q$	$q \geq O\sim(D \max(nT, D))$
本文 (推论 4.3)	$nT^2 \log D$	$n^2 T^2 D \log q$	$q \geq O\sim(nDT)$

表 3 一些文中的记号

$x, x_1, \dots, x_n$	变元
f	多项式
$\deg f$	f 的次数
$\#f$	f 的非零项数
D	f 的次数上界
T	f 的项数上界
n	f 的变元个数
$\mathbb{C}$	复数域
$\mathbb{F}_q$	有限域, 元素个数为 q
$\text{mod}(a, b)$	$a \bmod b$ 的余数, 且在 $\{0, 1, \dots, b-1\}$ 中, a 和 b 是正整数
$\mathcal{R}[\mathbb{X}] = \mathcal{R}[x_1, \dots, x_n]$	$\mathcal{R}$ 上的多项式环
M_f	f 中所有非零项的集合
$g(x) \bmod (x^p - 1)$	单元多项式 $g(x)$ 除以 $x^p - 1$ 的余数
$f_{(d,p)}$	$f(x, x^{\text{mod}(d,p)}, \dots, x^{\text{mod}(d^{n-1},p)})$
$f_{(d,p,k)}$	$f(x, \dots, x^{\text{mod}(d^{k-1},p)+p}, \dots, x^{\text{mod}(d^{n-1},p)})$
$f_{(d,p)}^{\text{mod}}$	$f_{(d,p)} \bmod (x^p - 1)$
$\mathcal{C}_{(d,p)}^f$	f 在 $f_{(d,p)}^{\text{mod}}$ 中的碰撞项的个数

1.2 与已有工作的比较

稀疏多项式可以由标准的黑盒模型给出, 也可以由更特殊的计算模型, 如 SLP 模型给出. 在 $\mathbb{Q}$ 或 $\mathbb{Z}$ 上, 由于除 0 和 ± 1 以外的点, D 次多项式在任何点的取值将具有 D 规模的大小, 从而, 在 $\mathbb{Q}$ 或 $\mathbb{Z}$ 上的标准黑盒模型上, 不存在位复杂度关于 $\log D$ 为多项式的插值算法. 甚至对于有限域 $\mathbb{F}_q$ 上的标准黑盒模型, 目前也未找到复杂度关于 $\log D$ 是多项式的插值算法. 但对于一些特殊模型, 例如, SLP 模型 [7–9, 12, 15–17]、有限精度黑盒模型 [2, 19–21] 和模黑盒模型 [22, 23], 存在复杂度关于 $\log D$ 是多项式的插值算法.

本文所提算法有三个主要步骤. 首先, 计算一组单变元多项式 $f_{(d,p)}^{\text{mod}}$, 并且证明如果 $f_{(d_0,p_0)}^{\text{mod}}$ 在其中具有最多的项数, 则 f 至少一半的项在 $f_{(d_0,p_0)}^{\text{mod}}$ 中不与 f 其他的项发生碰撞, 即在变元替换时不与其他项合并. 其次, 插值另外 n 个单变元多项式 $f(d_0, p_0, k)$, $k = 1, \dots, n$, 并从这些多项式中找出一个包含这些非碰撞项的集合. 最后, 给出一个给定单项式是否属于 f 的判定准则, 并使用这个准则从第二步中找到的单项式中提取出 f 的项. 经过以上步骤, 可找到 f 至少一半的项. 重复以上步骤即可以构造出 f .

在以上算法的第一步中, 我们把在基于基变换的 Kronecker 替换中的 d_0 和基于模变换的 Kronecker 替换中的 p_0 分别称作“ok”基和“ok”素数. 本文的主要贡献包括: 给出新的 Kronecker 替换、单项式是否属于 f 的检验准则, 以及寻找“ok”基和“ok”素数的确定性算法. 第 4 节给出的基于模变换的 Kronecker 替换是由本文第一次引入的. 第 3 节使用的基于基变换的 Kronecker 替换是由文献 [13] 引入的, 但如本节后面所介绍的, 本文的方法不同于文献 [13] 中的算法. 在本节的剩余部分中, 我们的算法将与文献 [7–9, 13, 15] 进行比较. 本文的工作基于以上工作并且受到这些工作的启发.

Garg 和 Schost [7] 引入了 (1.1) 中的替换 $f_{(D,p_j)}^{\text{mod}}$ 对 SLP 多项式进行插值. 他们证明了当 p_j 取

$O(T^2 \log D)$ 个不同的素数时, f 可从 $f_{(D,p_j)}^{\text{mod}}$ 中恢复. 我们的第二个算法类似于此方法, 但是我们的算法适用于更一般的黑盒模型, 并且具有以下差异. 第一, 对于黑盒多项式 f , 尽管 $f_{(D,p)}^{\text{mod}}$ 的次数小于 p , 但我们无法直接通过 p 计算 $f_{(D,p)}^{\text{mod}}$. 为了处理黑盒子多项式, 我们使用替换 $f_{(D,p)}$ 计算 $f_{(D,p)}^{\text{mod}} = f_{(D,p)} \bmod (x^p - 1)$. 第二, 我们只插值 $O(T \log D)$ 个不同的 $f_{(D,p_j)}^{\text{mod}}$, 其中 p_j 是素数. 第三, 我们给出判定一个单项式是否属于 f 的准则. 第四, 我们引入一种新的替换 $f_{(d,p,k)}$ 恢复多项式的次数.

替换 (1.2) 中的 $f_{(d,p)}$ 由 Klivans 和 Spielman [13] 引入并用于黑盒多项式插值. 不同于我们的 $f_{(d,p,k)}$, 他们用 $f_{(KS)} = f(q_1x, q_2x^{\text{mod}(d,p)}, \dots, q_nx^{\text{mod}(d^{n-1},p)})$ 恢复多项式的次数, 其中 q_i 是不同的素数. 我们的替换 $f_{(d,p,k)}$ 有以下优点: (1) 对于复数域, 在替换之后, 系数大小不会改变, 而 $f_{(KS)}$ 中的系数的大小增加了 D 倍; (2) 我们的算法适用于一般环, 而文献 [13] 中的算法需要存在一个元素 $\omega \in \mathcal{R}$, 使得 $\omega^i \neq 1, i = 1, 2, \dots, D$.

本文第 4 节给出的基于模变换的 Kronecker 替换可看作将文献 [9] 给出的插值方法从 SLP 模型推广到黑盒模型. 两者的主要的区别在于, 黑盒模型 $f_{(D,p)}^{\text{mod}}$ 是由 $f_{(D,p)} \bmod (x^p - 1)$ 计算而来. 因此, 计算中涉及的多项式的次数是 pD 而不是 p . Arnold 等 [16] 引入了 “ok” 素数的概念. 素数 p 称为 “ok” 的, 如果 f 中最多 $\frac{3}{8}T$ 的项在 $f_{(D,p)}^{\text{mod}}$ 中碰撞. 在文献 [16] 中, 计算 “ok” 素数的方法是一个概率算法. 本文给出的 “ok” 基与 “ok” 素数类似, 并且给出一种确定性方法计算 “ok” 基和 “ok” 素数. 此外, 文献 [15] 中使用的是随机 Kronecker 替换, 而我们的 Kronecker 型替换是确定性的.

对于有限域上的插值, Grigoriev 等 [18] 给出了第一个确定性算法. Prony 算法的复杂度主要由 $\mathbb{F}_q$ 中离散对数算法的代价决定. Ben-Or & Tiwari 算法和 Zippel 算法需要满足零点规避问题 [14, 24, 25], 因此较难推广为有限域上的确定性算法. Klivans-Spielman 算法需要计算因式分解, 因此只对满足如下条件的环有效: 包含一个元素 ω , 使得 $\omega^i \neq 1, i = 1, 2, \dots, D$. 我们的算法比现有算法更一般: 不需要满足零点规避问题, 也不需要求解离散对数问题和因式分解问题. 这主要是由于使用了 $f_{(d,p)}$ 和 $f_{(d,p,k)}$ 这些 Kronecker 类型替换.

2 三种 Kronecker 类型替换

在本文中, 令 $f = c_1m_1 + \dots + c_tm_t \in \mathcal{R}[\mathbb{X}]$ 是一个由黑盒 $\mathcal{B}_f$ 表示的多元多项式, 其中 $\mathcal{R}$ 是一个整环, $\mathbb{X} = \{x_1, \dots, x_n\}$ 是 n 个变元的集合, m_i ($i = 1, \dots, t$) 是不同的单项式, $c_i \in \mathcal{R}$. 记 $\#f = t$ 是 f 的项数, $\deg f$ 是 f 的总次数, $M_f = \{c_1m_1, \dots, c_tm_t\}$ 是 f 所有非零项的集合. $D, T \in \mathbb{N}$ 且满足 $D > \deg f, T \geq \#f$.

令 $d, p \in \mathbb{N}_{>0}$, x 是一个新的变元. 考虑 $\mathcal{R}[x]$ 中的单变元多项式:

$$f_{(d,p)} = f(x, x^{\text{mod}(d,p)}, \dots, x^{\text{mod}(d^{n-1},p)}), \quad (2.1)$$

$$f_{(d,p,k)} = f(x, \dots, x^{\text{mod}(d^{k-1},p)+p}, \dots, x^{\text{mod}(d^{n-1},p)}), \quad (2.2)$$

$$f_{(d,p)}^{\text{mod}} = f_{(d,p)} \bmod (x^p - 1) = f_{(d,p,k)} \bmod (x^p - 1), \quad (2.3)$$

其中 $k = 1, 2, \dots, n$. (2.1) 来自于改进的 Kronecker 替换 $x_i = x^{\text{mod}(d^{i-1},p)}, i = 1, 2, \dots, n$; (2.2) 来自另一个改进的 Kronecker 替换 $x_i = x^{\text{mod}(d^{i-1},p)}, i = 1, 2, \dots, n, i \neq k, x_k = x^{\text{mod}(d^{k-1},p)+p}$.

称 $cm \in M_f$ 在 $f_{(d,p)}^{\text{mod}}, f_{(d,p,k)}^{\text{mod}}$ 中碰撞, 如果存在 $aw \in M_f \setminus \{cm\}$ 使得 $m_{(d,p)}^{\text{mod}} = w_{(d,p)}^{\text{mod}}$ ($m_{(d,p)} = w_{(d,p)}$ 或者 $m_{(d,p,k)} = w_{(d,p,k)}$). 下面的结果是显然的.

引理 2.1 如果 $cm \in M_f$ 不在 $f_{(d,p)}^{\text{mod}}$ 中碰撞, 则 cm 也不在 $f_{(d,p)}$ 中碰撞, 并且 cm 也不在 $f_{(d,p,k)}$ ($k = 1, 2, \dots, n$) 中碰撞.

令

$$f_{(d,p)}^{\text{mod}} = a_1 x^{d_1} + a_2 x^{d_2} + \dots + a_r x^{d_r} \quad (d_1 < \dots < d_r). \quad (2.4)$$

由于 $f_{(d,p)}^{\text{mod}} = f_{(d,p)} \bmod (x^p - 1) = f_{(d,p,k)} \bmod (x^p - 1)$, $k = 1, 2, \dots, n$, $f_{(d,p)}$ 和 $f_{(d,p,k)}$ 可以写为

$$f_{(d,p)} = f_1 + f_2 + \dots + f_r + g, \quad (2.5)$$

$$f_{(d,p,k)} = f_{k,1} + f_{k,2} + \dots + f_{k,r} + g_k, \quad (2.6)$$

其中 $f_i \bmod (x^p - 1) = f_{k,i} \bmod (x^p - 1) = a_i x^{d_i}$ 和 $g \bmod (x^p - 1) = g_k \bmod (x^p - 1) = 0$ 对所有可能的 i 和 k 成立. 我们引入以下关键符号:

$$\begin{aligned} \text{TS}_{(d,p,D)}^f &= \left\{ a_i x_1^{e_{i,1}} \dots x_n^{e_{i,n}} \mid a_i \text{ 来自于 (2.4), 并且} \right. \\ &\quad \text{T}_1 : \text{在 (2.5) 和 (2.6) 中, } f_i = a_i x^{\gamma_i}, f_{k,i} = a_i x^{\beta_{k,i}}, k = 1, 2, \dots, n; \\ &\quad \text{T}_2 : e_{i,k} = \frac{\beta_{k,i} - \gamma_i}{p} \in \mathbb{N}, k = 1, 2, \dots, n; \\ &\quad \left. \text{T}_3 : \gamma_i = \sum_{k=1}^n e_{i,k} \bmod (d^{k-1}, p); \text{T}_4 : e_{i,1} + \dots + e_{i,n} < D \right\}. \end{aligned} \quad (2.7)$$

类似于文献 [9, 引理 5.3], 我们可以证明下面关于 $\text{TS}_{(d,p,D)}^f$ 的结果.

引理 2.2 令 $f = \sum_{i=1}^t c_i m_i \in \mathcal{R}[x_1, \dots, x_n]$, $d, p \in \mathbb{N}_{>0}$, $D > \deg(f)$. 如果 $c_i m_i$ 不在 $f_{(d,p)}^{\text{mod}}$ 中碰撞, 则 $c_i m_i \in \text{TS}_{(d,p,D)}^f$.

算法 1 用于计算集合 $\text{TS}_{(d,p,D)}^f$.

算法 1 TSTerms

输入: 单变元多项式 $f_{(d,p)}^{\text{mod}}, f_{(d,p)}, f_{(d,p,k)} \in \mathcal{R}[x]$, 其中 $k = 1, \dots, n$, $d \in \mathbb{N}_{>0}$, p 是素数, $D > \deg(f)$.

输出: $\text{TS}_{(d,p,D)}^f$.

第 1 步 将 $f_{(d,p)}^{\text{mod}}, f_{(d,p)}$ 和 $f_{(d,p,k)}$ 写成下面的形式:

$$\begin{aligned} f_{(d,p)}^{\text{mod}} &= a_1 x^{d_1} + \dots + a_r x^{d_r}, \\ f_{(d,p)} &= a_1 x^{\gamma_1} + \dots + a_\alpha x^{\gamma_\alpha} + g, \\ f_{(d,p,k)} &= a_1 x^{\beta_{k,1}} + \dots + a_\alpha x^{\beta_{k,\alpha}} + g_k \end{aligned}$$

使得对于 $i = 1, \dots, \alpha$, $k = 1, \dots, n$, 有 $\text{mod}(\gamma_i, p) = \text{mod}(\beta_{k,i}, p) = d_i$, $g_{(d,p)}^{\text{mod}} = (g_k)_{(d,p)}^{\text{mod}} = 0$.

第 2 步 令 $S = \{\}$. 对于 $i = 1, 2, \dots, \alpha$, 作如下操作:

- a:** 对于 $k = 1, 2, \dots, n$, 令 $e_{i,k} = \frac{\beta_{k,i} - \gamma_i}{p}$.
- b:** 如果 $e_{i,k}$ 不在 $\mathbb{N}$ 中, 则执行下一个循环;
- c:** 如果 $\sum_{k=1}^n e_{i,k} \bmod (d^{k-1}, p) \neq \gamma_i$, 则执行下一个循环;
- d:** 如果 $\sum_{k=1}^n e_{i,k} \geq D$, 则执行下一个循环;
- e:** 令 $S = S \cup \{a_i x_1^{e_{i,1}} \dots x_n^{e_{i,n}}\}$.

第 3 步 输出 S .

类似于文献 [9, 引理 5.5], 我们可以证明下面的引理.

引理 2.3 算法 1 需要 $O(nT)$ 次 $\mathcal{R}$ 环操作和 $O\sim(nT \log(pD))$ 次位操作.

3 基于基变换 Kronecker 替换的插值算法

该插值算法的基本想法是从 $f_{(d,p)}$ 中恢复 f , 其中 p 是一个固定素数, d 取足够多的整数值. 也就是说, 在 Kronecker 替换中, 基 d 是变化的.

3.1 项判定准则

本小节给出检验一个项 cm 是否属于一个多项式 f 的判定方法. 令 $\mathbb{F}_p$ 是一个有限域, 其中 p 是素数.

引理 3.1 令 $L_i = a_{i,1} + a_{i,2}x + \cdots + a_{i,n}x^{n-1} \in \mathbb{F}_p[x] \setminus \{0\}$, $i = 1, \dots, l$, p 是一个满足 $p \geq \max\{n, (n-1)l\}$ 的素数. 如果 δ 是一个满足 $(n-1)l \leq \delta \leq p$ 的整数, 那么在 $[1, \delta]$ 中存在至少 $\delta - (n-1)l$ 个整数 k 使得 $L_i(k) \neq 0$, $i = 1, 2, \dots, l$.

证明 对于每一个 L_i , 由于 $\deg(L_i) = n-1$, 因此, 在 $[1, \delta]$ 中存在至多 $n-1$ 个整数 k 使得 $L_i(k) = 0$. 由于我们有 l 个非零的函数 L_i , 因此, 在 $[1, \delta]$ 中, 存在至多 $(n-1)l$ 个整数 k 使得某个 $L_i(k) = 0$. 从而在 $[1, p]$ 中剩下的整数都不是 L_i ($i = 1, 2, \dots, l$) 的根. $\square$

引理 3.2 令 $f = \sum_{i=1}^t c_i m_i \in \mathcal{R}[\mathbb{X}]$, $T \geq \#f$, $D > \deg(f)$, $\delta_1 = (n-1)(T-1)$, p 是一个满足 $p \geq \max\{n, \delta_1, D\}$ 的素数, 则对任意满足 $\delta_1 \leq \delta \leq p$ 的整数 δ 和任意 $[1, t]$ 中的整数 i , 在 $[1, \delta]$ 中存在至少 $\delta - \delta_1$ 个整数 d , 使得 $c_i m_i$ 不在 $f_{(d,p)}^{\text{mod}}$ 中碰撞.

证明 如果 $t = 1$, 结论是显然的. 现在假定 $t \geq 2$. 只需考虑 $c_1 m_1$. 假设 $m_i = x_1^{e_{i,1}} x_2^{e_{i,2}} \cdots x_n^{e_{i,n}}$, $i = 1, 2, \dots, t$. 令 $L_s(x) = (e_{1,1} - e_{s,1}) + \cdots + (e_{1,n} - e_{s,n})x^{n-1} \bmod p$, 其中 $s = 2, 3, \dots, t$. 由于 m_1 和 m_s 是不同的单项式, 因此至少存在一个 k 使得 $e_{1,k} - e_{s,k} \neq 0$, $k \in \{1, 2, \dots, n\}$. 由于 $p \geq D$ 且 $|e_{1,i} - e_{s,i}| < D$, 因此所有 $L_s(x)$ 非零. 由于 $\deg(m_{1(d,p)}^{\text{mod}}) = \sum_{k=1}^n e_{1,k} d^{k-1} \bmod p$, $\deg(m_{s(d,p)}^{\text{mod}}) = \sum_{k=1}^n e_{s,k} d^{k-1} \bmod p$, 从而有 $m_{1(d,p)}^{\text{mod}} \neq m_{s(d,p)}^{\text{mod}}$. 由此可得, 如果 $d \in [1, \delta]$ 使得 $L_s(d) \neq 0$, 那么, $m_{1(d,p)}^{\text{mod}} \neq m_{s(d,p)}^{\text{mod}}$. 由引理 3.1 可知, 在 $[1, \delta]$ 中存在 $\geq \delta - (n-1)(T-1) \geq \delta - \delta_1$ 个整数使得所有 $L_s(x)$ ($s = 2, 3, \dots, t$) 非零, 即 m_1 不在 $f_{(d,p)}^{\text{mod}}$ 中碰撞. $\square$

现在给出一个验证 cm 是否属于 M_f 的判定准则.

定理 3.1 令 $f = \sum_{i=1}^t c_i m_i \in \mathcal{R}[\mathbb{X}]$, $T \geq \#f$, $D > \deg(f)$, $\delta_1 = (n-1)(T-1)$, $\delta_2 = (n-1)T$, p 是一个满足 $p \geq \max\{n, \delta_1 + \delta_2 + 1, D\}$ 的素数. 如果单项式 m 满足 $\deg(m) < D$, 那么, 项 $cm \in M_f$ 当且仅当在 $[1, \delta_1 + \delta_2 + 1]$ 中存在至少 $\delta_2 + 1$ 个整数 d , 使得 $\#(f - cm)_{(d,p)}^{\text{mod}} < \#f_{(d,p)}^{\text{mod}}$.

证明 令 $cm \in M_f$. 如果 d 是一个整数, 使得 cm 不在 $f_{(d,p)}^{\text{mod}}$ 中碰撞, 那么, $\#(f - cm)_{(d,p)}^{\text{mod}} = \#f_{(d,p)}^{\text{mod}} - 1$, 因此, $\#(f - cm)_{(d,p)}^{\text{mod}} < \#f_{(d,p)}^{\text{mod}}$. 根据引理 3.2, 令 $\delta = \delta_1 + \delta_2 + 1$, 则在 $[1, \delta]$ 中存在至少 $\delta_1 + \delta_2 + 1 - \delta_1 = \delta_2 + 1$ 个整数 d 使得 cm 不在 $f_{(d,p)}^{\text{mod}}$ 中碰撞. 所以存在至少 $\delta_2 + 1$ 个整数 d 使得 $\#(f - cm)_{(d,p)}^{\text{mod}} < \#f_{(d,p)}^{\text{mod}}$.

另一方面, 假设 $cm \notin M_f$. 我们证明在 $[1, \delta_1 + \delta_2 + 1]$ 中存在至多 δ_2 个整数 d 使得 $\#(f - cm)_{(d,p)}^{\text{mod}} < \#f_{(d,p)}^{\text{mod}}$. 考虑两种情形:

情形 1 m 不是 f 中的单项式. 那么 $f - cm$ 有至多 $T+1$ 项. 根据引理 3.2 知, 在 $[1, \delta_1 + \delta_2 + 1]$ 中存在至少 $\delta_1 + \delta_2 + 1 - \delta_2 = \delta_1 + 1$ 个整数 d 使得 cm 不在 $(f - cm)_{(d,p)}^{\text{mod}}$ 中碰撞. 所以至少存在 $\delta_1 + 1$

个整数 d 使得 $\#(f - cm)_{(d,p)}^{\text{mod}} > \#f_{(d,p)}^{\text{mod}}$. 所以存在至多 $\delta_1 + \delta_2 + 1 - (\delta_1 + 1) = \delta_2$ 个整数 d 使得 $\#(f - cm)_{(d,p)}^{\text{mod}} < \#f_{(d,p)}^{\text{mod}}$.

情形 2 m 是 f 中的单项式, 但是 cm 不是 f 的项. 那么 $f - cm$ 与 f 的项数相同. 令 f 中单项式为 m 的项是 $c_1 m$. 根据引理 3.2 知, 存在至少 $\delta_1 + \delta_2 + 1 - \delta_1 = \delta_2 + 1$ 个整数 d 使得 $(c_1 - c)m$ 不在 $(f - cm)_{(d,p)}^{\text{mod}}$ 中碰撞. 所以存在至少 $\delta_2 + 1$ 个整数 d 使得 $\#(f - cm)_{(d,p)}^{\text{mod}} = \#f_{(d,p)}^{\text{mod}}$. 由于 $\delta_1 + \delta_2 + 1 - (\delta_2 + 1) = \delta_1 < \delta_2$, 所以存在至多 δ_2 个整数 d 使得 $\#(f - cm)_{(d,p)}^{\text{mod}} < \#f_{(d,p)}^{\text{mod}}$. $\square$

3.2 计算 “ok” 基

本小节将展示如何去找到一个 “ok” 基 d 使得 f 至少一半的单项式不在 $f_{(d,p)}^{\text{mod}}$ 中碰撞, 其中 p 是一个固定的素数.

引理 3.3 ^[10] 令 B_i ($i = 1, 2, \dots, s$) 是整数构成的有限非空集合, a_i ($i = 1, 2, \dots, t$) 是所有在 $\bigcup_{j=1}^s B_j$ 中的不同元素, 且任意 B_i 两两相交为空. 如果 c 是满足 $\#B_i \geq 2$ 的集合中的所有不同元素个数的总和, 那么 $t - c \leq s$, 并且对于在 $[t - c, s]$ 中的任意整数 s_1 , 有 $(t - s_1) \leq c \leq 2(t - s_1)$.

记 $\mathcal{C}_{(d,p)}^f$ 是 f 在 $f_{(d,p)}^{\text{mod}}$ 中的碰撞项的个数. 我们有下面的引理:

引理 3.4 令 $f \in \mathcal{R}[\mathbb{X}]$, $\#f_{(d_0,p)}^{\text{mod}} = s_0$, $\#f_{(d,p)}^{\text{mod}} = s$. 如果 $s_0 \geq s$, 那么 $\mathcal{C}_{(d_0,p)}^f \leq 2\mathcal{C}_{(d,p)}^f$.

证明 假设 $f_{(d_0,p)}^{\text{mod}} = a_1 x^{d_1} + \dots + a_{s_0} x^{d_{s_0}}$, 且当 $i \neq j$ 时, $d_i \neq d_j$. 令 $f = f_1 + \dots + f_{s_0} + g$, 其中 $f_{i(d_0,p)}^{\text{mod}} = a_i x^{d_i}$, $i = 1, \dots, s_0$, $g_{(d_0,p)}^{\text{mod}} = 0$. 令 B_i ($i = 1, \dots, s_0$) 是 f_i 中所有项的集合, B_0 是 g 中所有项的集合. 根据引理 3.3, 有 $(t - s_0) < \mathcal{C}_{(d_0,p)}^f \leq 2(t - s_0)$. 同理, $(t - s) \leq \mathcal{C}_{(d,p)}^f \leq 2(t - s)$. 从而, $\mathcal{C}_{(d_0,p)}^f \leq 2(t - s_0) \leq 2(t - s) \leq 2\mathcal{C}_{(d,p)}^f$. 引理已证. $\square$

引理 3.5 令 $f = \sum_{i=1}^t c_i m_i \in \mathcal{R}[\mathbb{X}]$, $m_i = x_1^{e_{i,1}} \dots x_n^{e_{i,n}}$, $D > \deg(f)$, p 是一个满足 $p \geq \max\{n, D\}$ 的素数, $A_{u,v} = \sum_{j=1}^n (e_{u,j} - e_{v,j}) x^{j-1} \bmod p$, 其中 $u, v \in \{1, \dots, t\}$, 并且 $u < v$. 如果 $\mathcal{C}_{(d_0,p)}^f = s$ 且 d_0 是一个在 $[1, p]$ 的整数, 那么存在至少 $\lceil \frac{s}{2} \rceil$ 个 $A_{u,v}$ 使得 $A_{u,v}(d_0) = 0$.

证明 $c_{j_1} m_{j_1} + \dots + c_{j_i} m_{j_i}$ 称为一个 f 在 $f_{(d_0,p)}^{\text{mod}}$ 中的碰撞块, 如果对于任意 $u, v \in \{j_1, \dots, j_i\}$, 有 $m_{u(d_0,p)}^{\text{mod}} = m_{v(d_0,p)}^{\text{mod}}$. 假设 $c_{j_1} m_{j_1} + \dots + c_{j_i} m_{j_i}$ 是一个碰撞块. 对于任意的 $u, v \in \{j_1, \dots, j_i\}$, $u < v$, 有 $m_{u(d_0,p)}^{\text{mod}} = m_{v(d_0,p)}^{\text{mod}}$. 所以, $(e_{u,1} + \dots + e_{u,n} d_0^{n-1}) \bmod p = (e_{v,1} + \dots + e_{v,n} d_0^{n-1}) \bmod p$, 由此推出 d_0 是 $A_{u,v}$ 的根. 令 n_i 是在 $f_{(d_0,p)}^{\text{mod}}$ 中含 i 项的碰撞块个数, 则存在 $C_i^2 = \frac{i(i-1)}{2}$ 个这样的对 (u, v) 与 n_i 个这样的碰撞块. 令 $K = \sum_{i=1}^t \frac{1}{2} (i^2 - i) n_i$, 从而存在 K 个不同的 $A_{u,v}$ 有根 d_0 . 现在给出 K 的一个下界. 首先有 $t = \sum_{i=1}^t i n_i$, $s = \sum_{i=2}^t i n_i$. 所以, $K = \sum_{i=1}^t \frac{1}{2} (i^2 - i) n_i = \frac{1}{2} \sum_{i=1}^t i^2 n_i - \frac{1}{2} \sum_{i=1}^t i n_i = \frac{1}{2} \sum_{i=1}^t i^2 n_i - \frac{1}{2} t = \frac{1}{2} n_1 + \frac{1}{2} \sum_{i=2}^t i^2 n_i - \frac{1}{2} t \geq \frac{1}{2} n_1 + t - n_1 - \frac{1}{2} t = \frac{1}{2} t - \frac{1}{2} n_1 = \frac{1}{2} s$. 因为 K 是一个整数, 所以, $K \geq \lceil \frac{1}{2} s \rceil$. $\square$

下面定理给出一个计算 “ok” 基的方法.

定理 3.2 令 $f = \sum_{i=1}^t c_i m_i \in \mathcal{R}[\mathbb{X}]$, $T \geq \#f$, $D > \deg f$, $\delta_1 = (n-1)(T-1)$, p 是一个满足 $p \geq \max\{n, 4\delta_1 + 1, D\}$ 的素数. 令 d_0 是一个在 $[1, 4\delta_1 + 1]$ 中的整数, 且对于所有的 $d = 1, \dots, 4\delta_1 + 1$, 均有 $\#f_{(d_0,p)}^{\text{mod}} \geq \#f_{(d,p)}^{\text{mod}}$, 那么, f 至少包含 $\lceil \frac{t}{2} \rceil$ 个不在 $f_{(d_0,p)}^{\text{mod}}$ 中碰撞的单项式.

证明 如果 $t = 1$, 结论显然. 假设 $t \geq 2$. 首先证明结论: 在 $[1, 4\delta_1 + 1]$ 中存在至少一个整数 d 使得 $\mathcal{C}_{(d,p)}^f < \frac{t}{4}$. 我们反证该结论. 假设对于 $d = 1, \dots, 4\delta_1 + 1$, 均有 $\mathcal{C}_{(d,p)}^f \geq \frac{t}{4}$. 那么根据引理 3.5 知, 存在 $\lceil \frac{1}{2} \mathcal{C}_{(d,p)}^f \rceil$ 个多项式 $A_{u,v}$, $u, v \in \{1, \dots, t\}$, $u < v$ 使得 $A_{u,v}(d) = 0$. 由于 $d = 1, \dots, 4\delta_1 + 1$ 是在 $\mathbb{F}_p$ 中不同的元素, 那么所有的 $A_{u,v}$, $u, v \in \{1, \dots, t\}$, $u < v$ 至少有 $\sum_{d=1}^{4\delta_1+1} \lceil \frac{1}{2} \mathcal{C}_{(d,p)}^f \rceil$ 个根. 注意到 $\sum_{d=1}^{4\delta_1+1} \lceil \frac{1}{2} \mathcal{C}_{(d,p)}^f \rceil > 4\delta_1 \lceil \frac{1}{2} \cdot \frac{t}{4} \rceil \geq \frac{1}{2} \delta_1 t = \frac{1}{2} (n-1)(T-1)t$, 这与 $\{A_{u,v}, u, v \in \{1, 2, \dots, t\}, u < v\}$ 的次数和

至多是 $\frac{1}{2}t(t-1)(n-1)$ 矛盾. 结论已证. 将引理 3.4 应用于刚刚证明的结论, 我们有 $\mathcal{C}_{(d_0,p)}^f \leq 2\mathcal{C}_{(d,p)}^f$, $d = 1, \dots, 4\delta_1 + 1$, 所以, $\mathcal{C}_{(d_0,p)}^f < 2 \cdot \frac{t}{4} = \frac{t}{2}$. 从而, $f_{(d_0,p)}^{\text{mod}}$ 中不碰撞的项至少大于 $t - \frac{t}{2} = \frac{t}{2}$, 也就是大于等于 $\lceil \frac{t}{2} \rceil$. $\square$

3.3 将多变元插值约化到单变元插值

本小节给出一个约化算法, 将多元多项式的插值问题约化为单元多项式插值. 在该算法中, 假设单元多项式插值算法是存在的.

算法 2 MIPolyBase

输入: 一个计算 $f \in \mathcal{R}[x_1, \dots, x_n]$ 的黑盒程序 $\mathcal{B}_f$, $T \geq \#f$, $D > \deg f$.

输出: f 的具体形式.

第 1 步 令 $\delta_1 = (n-1)(T-1)$, $\delta_2 = (n-1)T$, $N = \max\{4\delta_1 + 1, \delta_1 + \delta_2 + 1\}$. 计算一个素数 p 使得 $p \geq \max\{n, N, D\}$.

第 2 步 对于 $d = 1, 2, \dots, N$, 使用单变元多项式插值算法计算 $f_{(d,p)}$. 令 $f_d = f_{(d,p)}$, $f_d^{\text{mod}} = f_{(d,p)}^{\text{mod}}$.

第 3 步 令 $\alpha = \max\{\#f_d^{\text{mod}} \mid d = 1, 2, \dots, N\}$, d_0 满足 $\#f_{d_0}^{\text{mod}} = \alpha$. 令 $h = 0$.

第 4 步 当 $\alpha \neq 0$ 时, 重复如下操作:

a: 对于 $k = 1, 2, \dots, n$, 通过单变元多项式插值算法计算 $f_{(d_0,p,k)}$, 并令 $g_k = f_{(d_0,p,k)} - h_{(d_0,p,k)}$.

b: 令 $\text{TS}_{(d_0,p,D)}^{f-h} = \text{TS}_{(d_0,p,D)}(f_{d_0}^{\text{mod}}, f_{d_0}, g_1, \dots, g_n, d_0, p, D)$.

c: 令 $s = 0$. 对于任意的 $u \in \text{TS}_{(d_0,p,D)}^f$, 如果

$$\#\{d \mid \#(f_d^{\text{mod}} - u_{(d,p)}^{\text{mod}}) < \#f_d^{\text{mod}}, d = 1, \dots, \delta_1 + \delta_2 + 1\} \geq \delta_2 + 1,$$

则 $s := s + u$.

d: 令 $h = h + s$, $T = T - \#s$, $\delta_1 = (n-1)(T-1)$, $\delta_2 = (n-1)T$, $N = \max\{4\delta_1 + 1, \delta_1 + \delta_2 + 1\}$.

e: 对于 $d = 1, \dots, N$, 令 $f_d = f_d - s_{(d,p)}$, $f_d^{\text{mod}} = f_d^{\text{mod}} - s_{(d,p)}^{\text{mod}}$.

f: 令 $\alpha = \max\{\#f_d^{\text{mod}} \mid d = 1, \dots, N\}$, 且 d_0 满足 $\#f_{d_0}^{\text{mod}} = \alpha$.

第 5 步 输出 h .

定理 3.3 算法 2 正确. 该算法需要 $O(nT)$ 次的单元多项式插值, 这些单元多项式次数小于 $O(D \max\{O(nT), O(D)\})$, 项数小于 T . 此外, 它还需要 $O(nT^2)$ 次的 $\mathcal{R}$ 环操作和 $O(n^2T^2 \log D)$ 次位操作.

证明 根据定理 3.2 和引理 2.2, 在第 4 步的 **b** 中, f 中至少一半的单项式在 $\text{TS}_{(d_0,p,D)}^f$ 中. 在第 4 步的 **c** 中, 由定理 3.1, 可以从 $\text{TS}_{(d_0,p,D)}^f$ 中选出 f 中至少一半的单项式. 从而, 每次循环第 4 步后, 至少得到 f 中的一半的单项式. 所以最多循环 $\log_2 T$ 次第 4 步, 可以恢复 f . 所以, 算法的正确性已证.

现在分析其计算复杂度. 在第 2 步中, 调用 N 次单变元多项式插值; 在第 4 步的 **a** 中, 至多调用 $n \log_2 t$ 次单变元插值. 注意到 $\deg(f_{(d,p)}) \leq D(p-1)$ 和 $\deg f_{(d,p,k)} \leq 2D(p-1)$. 由于 $p = \max\{O(nT), O(D)\}$ 且 $\#f \leq T$, 定理的第一部分已证.

在第 2 步中, 计算一个 f_d^{mod} 需要 $O(t)$ 次环操作和 $O(t \log(pD))$ 次位操作. 因为其总共有 N 个多项式 f_d , 所以, 复杂度是 $O(nT^2)$ 次的环操作和 $O(nT^2 \log D)$ 次位操作.

在第 3 步中, 由于 $N = O(nT)$ 且 $f_{(d_0,p)}^{\text{mod}}$ 不超过 T 项, 所以需要 $O(nT^2)$ 次位操作.

对于第 4 步, 首先分析一次循环的复杂度. 在 **b** 中, 根据引理 2.3 可知, 复杂度是 $O(nT)$ 次的环操作和 $O(nT \log D)$ 次位操作. 在 **c** 中, 由于每个检查需要比较 $\delta_1 + \delta_2 + 1$ 个多项式和用 $\delta_1 + \delta_2 + 1$

个项的替换, 所以, 复杂度是 $O(\#TS_{(d_0,p,D)}^f n(\delta_1 + \delta_2) \log(pD) + \#TS_{(d_0,p,D)}^f (\delta_1 + \delta_2) \log T \log(pD))$ 次位操作和 $O(\#TS_{(d_0,p,D)}^f (\delta_1 + \delta_2))$ 次环操作. 由于 $\#TS_{(d_0,p,D)}^f \leq T$, 所以, 复杂度是 $O(nT^2)$ 次的环操作和 $O(n^2 T^2 \log D)$ 次位操作. 在 **e** 中, 需要 $n(\#s)$ 次的操作去获得 $s_{(d,p)}$ 和 $s_{(d,p)}^{\text{mod}}$. 由于我们需要更新 N 个多项式, 所以, 复杂度是 $O(n(\#s)N \log(pD))$ 次位操作和 $O(\#sN)$ 次环操作. 由于每次循环得到 f 的至少一半的项, 所以至多需要 $\log_2 t$ 次循环. 由于 $\#s$ 的和是 t , 所以, 第 4 步的总的复杂度是 $O(nT^2)$ 次环操作和 $O(n^2 T^2 \log D)$ 次位操作. $\square$

3.4 多元多项式插值

算法 2 将多元多项式插值约化为单元多项式插值. 本小节结合算法 2 和三种单元多项式插值算法给出多元插值算法, 并且将它们作为定理 3.3 的推论.

对于一个 D 次单变元多项式 f , Lagrange 插值算法适用于任何含有多于 $D+1$ 个元素的整环. 该算法需要对 f 进行 $O(D)$ 次赋值和 $O(D)$ 次环操作^[26]. 结合定理 3.3 和 Lagrange 算法, 我们有以下结果.

推论 3.1 令 $\mathcal{R}$ 是一个含有超过 $O(D \max(nT, D))$ 个元素的整环. 如果使用 Lagrange 算法计算单变元插值, 那么, 算法 2 需要 $O(nTD \max(nT, D))$ 次对黑盒 $\mathcal{B}_f$ 的查询和 $O(nTD \max(nT, D))$ 次 $\mathcal{R}$ 算术操作, 以及少量的位操作.

如果 $\mathcal{R}$ 是复数域, 那么, 我们可用 Ben-Or & Tiwari 算法计算单变元多项式插值, 其复杂度被求根算法^[1,3] 所控制. 文献 [27, 引理 2.3] 证明了对于单变元多项式, 根可以通过分解辅助多项式的系数而得到. 因此, 单变元 Ben-Or & Tiwari 算法需要 $2T$ 次赋值, 其复杂度为 $O(T \log^2 D)$ 次域操作. 由于 $T \leq D$, 因此, 这个复杂度比 Lagrange 算法复杂度更好. 结合定理 3.3, 假设系数的大小是固定的, 我们给出如下结果.

推论 3.2 若 $\mathcal{R} = \mathbb{C}$ 且 Ben-Or & Tiwari 算法被用于单元多项式插值, 则算法 2 需要 $O(nT^2 \log D)$ 次对黑盒 $\mathcal{B}_f$ 的查询和 $O(nT^3 D \max(nT, D))$ 次位操作.

证明 由于 $\#f_{(d,p)}, \#f_{(d,p,k)} \leq T$ 且 $\deg(f_{(d,p)})$ 和 $\deg(f_{(d,p,k)})$ 的大小是 $O(D \max(nT, D))$, 所以, 插值单变元多项式的算术复杂度是 $O(nT^2 \log^2 D)$. 由于我们在 $2^0, \dots, 2^{2T-1}$ 点赋值, 从而, 数据长度是 $O(TD \max(nT, D))$, 所以, 算法 2 的位复杂度是 $O(nT^3 D \max(nT, D))$. $\square$

如果 $\mathcal{R} = \mathbb{F}_q$, 则单变元 Ben-Or & Tiwari 算法可以修改为 $\mathbb{F}_q$ 上的确定性算法. 它需要 $2T$ 个赋值和 $O(D \log q)$ 次的位操作^[10]. Lagrange 算法与它有相同的复杂度 $O(D \log q)$, 但是 Lagrange 算法需要 $O(D)$ 次赋值. 基于这种修改后的 Ben-Or & Tiwari 算法, 我们有下面的推论:

推论 3.3 如果 $\mathcal{R} = \mathbb{F}_q$ 且使用修改后的 Ben-Or & Tiwari 算法计算单变元多项式插值, 则算法 2 需要 $O(nT^2)$ 次对黑盒 $\mathcal{B}_f$ 的查询和 $O(nTD \max(nT, D) \log q)$ 次的额外的位操作.

证明 由于 $f_{(d,p)}$ 和 $f_{(d,p,k)}$ 的次数是 $O(D \max(nT, D))$, $\#f_{(d,p)}, \#f_{(d,p,k)} \leq T$, 由文献 [10] 和定理 3.3 可以得到位复杂度. $\square$

在这三个算法中, 推论 3.1 中的算法作用于更一般的环, 但它具有较高的复杂度. 推论 3.2 和 3.3 中的两个算法都使用 Ben-Or & Tiwari 算法进行单变元插值, 它们适用于不同的系数环.

4 基于模变换 Kronecker 替换的插值算法

本节给出一种从 $f_{(D,p)}$ 中恢复 f 的算法, 其中 D 是 f 的次数上界, p 取充分多个素数. 在该算法

中, 模 p 在 Kronecker 替换中是变化的. 该算法与第 3 节给出的算法相似, 不同之处是, 我们需要给出新的单项式判定准则以及类计算 “ok” 素数的算法.

4.1 单项式的验证准则和 “ok” 素数

首先证明下面的引理.

引理 4.1 令 $f = \sum_{i=1}^t c_i m_i$, $T \geq \#f$, $D > \deg f$, N_1 是使得 $p_1 p_2 \cdots p_{N_1} \geq D^{n(T-1)}$ 成立的最小整数, 其中 p_i 是第 i 个素数. 对于每一个 $cm \in M_f$, 存在最多 $N_1 - 1$ 个素数 p , 使得 cm 在 $f_{(D,p)}^{\text{mod}}$ 中碰撞.

证明 如果 $t = 1$, 那么 $N_1 \geq 1$, 结论显然成立. 假设 $t \geq 2$. 只需证明在任意 N_1 个不同的素数 $q_1, q_2, \dots, q_{N_1}$ 中, 存在至少一个 q_j , 使得 cm 不在 $f_{(D,q_j)}^{\text{mod}}$ 中碰撞. 假设 $m_i = x_1^{e_{i,1}} \cdots x_n^{e_{i,n}}$, $i = 1, \dots, t$. 只需考虑 $c_1 m_1$ 是否在 $f_{(D,q_j)}^{\text{mod}}$ 中碰撞. 我们用反证法证明: 假设对于每一个 $q_j, j = 1, \dots, N_1$, $c_1 m_1$ 都在 $f_{(D,q_j)}^{\text{mod}}$ 中碰撞. 令 $B = \prod_{s=2}^t (\sum_{i=1}^n (e_{1,i} - e_{s,i}) D^{i-1})$. 首先, 我们证明如果 $c_1 m_1$ 在 $f_{(D,q_j)}^{\text{mod}}$ 中碰撞, 那么, $\text{mod}(B, q_j) = 0$. 由于 $c_1 m_1$ 在 $f_{(D,q_j)}^{\text{mod}}$ 中碰撞, 不失一般性, 假设 $m_{1(D,q_j)}^{\text{mod}} = m_{2(D,q_j)}^{\text{mod}}$. 那么, $0 = \deg(m_{1(D,q_j)}^{\text{mod}}) - \deg(m_{2(D,q_j)}^{\text{mod}}) = \text{mod}(\sum_{i=1}^n e_{1,i} D^{i-1}, q_j) - \text{mod}(\sum_{i=1}^n e_{2,i} D^{i-1}, q_j)$. 所以, $\text{mod}(\sum_{i=1}^n (e_{1,i} - e_{2,i}) D^{i-1}, q_j) = 0$, 也即 $\text{mod}(B, q_j) = 0$. 因为 $q_1, \dots, q_{N_1}$ 是不同的素数, 所以, $\prod_{j=1}^{N_1} q_j$ 整除 B . 注意到 $|\sum_{i=1}^n (e_{1,i} - e_{s,i}) D^{i-1}| \leq (D-1)(\sum_{i=1}^n D^{i-1}) = D^n - 1$. 所以, $|B| = |\prod_{s=2}^t \sum_{i=1}^n (e_{1,i} - e_{s,i}) D^{i-1}| \leq (D^n - 1)^{t-1}$. 因此, $\prod_{j=1}^{N_1} q_j \geq \prod_{j=1}^{N_1} p_j \geq D^{n(T-1)} > (D^n - 1)^{T-1} \geq |B|$. 这与事实 $\prod_{j=1}^{N_1} q_j$ 整除 B 矛盾. 所以至少存在一个 $q_j (j = 1, \dots, N_1)$ 不整除 B . 不失一般性, 假设 q_1 不整除 B . 那么, $\text{mod}(\sum_{i=1}^n e_{1,i} D^{i-1} - \sum_{i=1}^n e_{s,i} D^{i-1}, q_1) \neq 0$. 所以, 对于 $s = 2, 3, \dots, t$, 均有 $\text{mod}(\sum_{i=1}^n e_{1,i} D^{i-1}, q_1) \neq \text{mod}(\sum_{i=1}^n e_{s,i} D^{i-1}, q_1)$. 所以, $c_1 m_1$ 不在 $f_{(D,q_1)}^{\text{mod}}$ 中碰撞. 引理已证. $\square$

下面给出一个判定 $cm \in M_f$ 的新方法.

定理 4.1 令 $f = \sum_{i=1}^t c_i m_i$, $T \geq \#f$, $D > \deg f$, N_1 是使得 $p_1 p_2 \cdots p_{N_1} \geq D^{n(T-1)}$ 成立的最小整数, N_2 是使得 $p_1 p_2 \cdots p_{N_2} \geq D^{nT}$ 成立的最小整数, 其中 p_i 是第 i 个素数. 如果单项式 cm 满足 $\deg(m) < D$, 那么, $cm \in M_f$ 当且仅当在 $[1, N_1 + N_2 - 1]$ 中存在至少 N_2 个整数 j 使得 $\#(f - cm)_{(D,p_j)}^{\text{mod}} < \#f_{(D,p_j)}^{\text{mod}}$.

证明 首先假定 $cm \in M_f$. 若 p_j 是一个素数, 使得 cm 不在 $f_{(D,p_j)}^{\text{mod}}$ 中碰撞, 则 $\#(f - cm)_{(D,p_j)}^{\text{mod}} = \#f_{(D,p_j)}^{\text{mod}} - 1$, 所以, $\#(f - cm)_{(D,p_j)}^{\text{mod}} < \#f_{(D,p_j)}^{\text{mod}}$. 根据引理 4.1, 存在至多 $N_1 - 1$ 个素数使得 cm 在 f 中碰撞. 因为 $N_1 + N_2 - 1 - (N_1 - 1) = N_2$, 所以在 $p_1, \dots, p_{N_1+N_2-1}$ 中存在至少 N_2 个素数使得 $\#(f - cm)_{(D,p_j)}^{\text{mod}} < \#f_{(D,p_j)}^{\text{mod}}$. 现在证明另一方向. 假设 $cm \notin M_f$. 考虑两种情形:

情形 1 m 不是 f 的单项式. 在这种情形下, $f - cm$ 有至多 $T + 1$ 项. 根据引理 4.1 可知, 在 $[1, N_1 + N_2 - 1]$ 中存在至多 $N_2 - 1$ 个整数 j 使得 cm 在 $(f - cm)_{(D,p_j)}^{\text{mod}}$ 中碰撞. 由于 $N_1 + N_2 - 1 - (N_2 - 1) = N_1$, 因此, 在 $p_1, \dots, p_{N_1+N_2-1}$ 中存在至少 N_1 个素数使得 $\#(f - cm)_{(D,p_j)}^{\text{mod}} > \#f_{(D,p_j)}^{\text{mod}}$. 所以存在至多 $N_1 + N_2 - 1 - N_1 = N_2 - 1$ 个素数使得 $\#(f - cm)_{(D,p_j)}^{\text{mod}} < \#f_{(D,p_j)}^{\text{mod}}$.

情形 2 m 是 f 中单项式, 但 cm 不是 f 中的单项式. 在这种情形下, $f - cm$ 与 f 有同样的项数. 假设 $c_1 m \in M_f$. 根据引理 4.1 可知, 存在至多 $N_1 - 1$ 个素数使得 $(c_1 - c)m$ 在 $(f - cm)_{(D,p_j)}^{\text{mod}}$ 中碰撞. 由于 $N_1 + N_2 - 1 - (N_1 - 1) = N_2$, 因此, 在 $p_1, \dots, p_{N_1+N_2-1}$ 中存在至少 N_2 个素数使得 $\#(f - cm)_{(D,p_j)}^{\text{mod}} = \#f_{(D,p_j)}^{\text{mod}}$. 从而存在至多 $N_1 + N_2 - 1 - N_2 = N_1 - 1 \leq N_2 - 1$ 个素数使得 $\#(f - cm)_{(D,p_j)}^{\text{mod}} < \#f_{(D,p_j)}^{\text{mod}}$. $\square$

如果 f 的项至少一半不在 $f_{(D,p)}^{\text{mod}}$ 中碰撞, 那么这样的素数 p 称作 “ok” 素数.

引理 4.2^[9] 令 $f = \sum_{i=1}^t c_i m_i \in \mathcal{R}[\mathbb{X}]$, $m_i = x_1^{e_{i,1}} \cdots x_n^{e_{i,n}}$, $D > \deg(f)$, $A = \prod_{i,j \in [1,t]}^{i < j} \sum_{k=1}^n (e_{i,k} - e_{j,k}) D^{k-1}$, p 是一个素数. 如果 $\mathcal{C}_{(D,p)}^f = s$, 那么, $p^{\lceil \frac{s}{2} \rceil}$ 整除 A .

下面给出一个计算 “ok” 素数的方法.

定理 4.2 令 $f = \sum_{i=1}^t c_i m_i \in \mathcal{R}[\mathbb{X}]$, $T \geq \#f$, $D > \deg f$, N_3 是使得 $p_1 \cdots p_{N_3} \geq D^{4n(T-1)}$ 成立的最小的整数, 其中 p_i 是第 i 个素数. 令 p_{j_0} 满足 $\#f_{(D,p_{j_0})}^{\text{mod}} \geq \#f_{(D,p_j)}^{\text{mod}}$, $j = 1, \dots, N_3$, 则 p_{j_0} 是一个 “ok” 素数.

证明 首先证明结论: 在 $p_1, \dots, p_{N_3}$ 中至少有一个 p_j 使得 $\mathcal{C}_{(D,p_j)}^f < \frac{t}{4}$. 假设此结论不正确, 即对于 $j = 1, \dots, N_3$, 均有 $\mathcal{C}_{(D,p_j)}^f \geq \frac{t}{4}$. 那么根据引理 4.2 可知, $p_j^{\lceil \frac{1}{2} \mathcal{C}_{(D,p_j)}^f \rceil}$ 整除 A . 由于 p_j ($j = 1, \dots, N_3$) 是不同的素数, 所以, $\prod_{j=1}^{N_3} p_j^{\lceil \frac{1}{2} \mathcal{C}_{(D,p_j)}^f \rceil}$ 整除 A . 由假设 $p_1 \cdots p_{N_3} \geq D^{4n(T-1)}$ 可推得 $\prod_{j=1}^{N_3} p_j^{\lceil \frac{1}{2} \mathcal{C}_{(D,p_j)}^f \rceil} \geq D^{\frac{1}{2} n t (T-1)}$, 这与事实 $A \leq (D^n - 1)^{\frac{t(t-1)}{2}}$ 矛盾. 我们证明了这个结论. 根据引理 3.4, 有 $\mathcal{C}_{(D,p_{j_0})}^f \leq 2\mathcal{C}_{(D,p_j)}^f$, $j = 1, \dots, N_3$, 所以, $\mathcal{C}_{(D,p_{j_0})}^f < \frac{t}{2}$. 从而, f 不在 $f_{(D,p_{j_0})}^{\text{mod}}$ 中碰撞的单项式的个数大于 $t - \frac{t}{2} = \frac{t}{2}$. 定理正确. $\square$

4.2 从多变元多项式插值约化到单变元多项式插值

基于定理 4.1 和 4.2, 现在给出基于模变换 Kronecker 替换的插值算法, 即算法 3.

算法 3 MIPoly2

输入: 一个黑盒子多项式 $f \in \mathcal{R}[x_1, \dots, x_n]$, $T \geq \#f$, $D > \deg f$.

输出: f 的具体形式.

第 1 步 令 $K = \max\{4, \lceil n(T-1) \log_2 D \rceil + \lceil nT \log_2 D \rceil, 4\lceil nT \log_2 D \rceil\}$. 计算前 K 个素数 $p_1, \dots, p_K$.

第 2 步 根据定理 4.1 和 4.2 计算 N_1 , N_2 和 N_3 . 令 $N = \max\{N_1 + N_2 - 1, N_3\}$ ($K \geq N$).

第 3 步 对于 $j = 1, \dots, N$, 使用单变元插值算法计算 $f_{(D,p_j)}$. 令 $f_j = f_{(D,p_j)}$, $f_j^{\text{mod}} = f_{(D,p_j)}^{\text{mod}}$.

第 4 步 令 $\alpha = \max\{\#f_j^{\text{mod}} \mid j = 1, 2, \dots, N\}$, j_0 满足 $\#f_{j_0}^{\text{mod}} = \alpha$. 令 $h = 0$.

第 5 步 当 $\alpha \neq 0$ 时, 作如下操作:

a: 对于 $k = 1, 2, \dots, n$, 通过单变元插值算法计算 $f_{(D,p_{j_0},k)}$. 令 $g_k = f_{(D,p_{j_0},k)} - h_{(D,p_{j_0},k)}$.

b: 计算 $\text{TS}_{(D,p_{j_0},D)}^{f-h} = \text{TS}_{\text{Terms}}(f_{j_0}^{\text{mod}}, f_{j_0}, g_1, \dots, g_n, D, p_{j_0}, D)$.

c: 令 $s = 0$. 对于每个 $u \in \text{TS}_{(D,p_{j_0},D)}^f$, 如果

$$\#\{j \mid \#(f_j^{\text{mod}} - u_{(D,p_j)}^{\text{mod}}) < \#(f_j^{\text{mod}}), j = 1, \dots, N_1 + N_2 - 1\} \geq N_2,$$

那么令 $s = s + u$.

d: 令 $h = h + s$, $T = T - \#s$, 对新的 T 更新 N_1 , N_2 和 N_3 . 令 $N = \max\{N_3, N_1 + N_2 - 1\}$.

e: 对于 $j = 1, 2, \dots, N$, 令 $f_j = f_j - s_{(D,p_j)}$, $f_j^{\text{mod}} = f_j^{\text{mod}} - s_{(D,p_j)}^{\text{mod}}$.

f: 令 $\alpha = \max\{\#f_j^{\text{mod}} \mid j = 1, 2, \dots, N\}$, j_0 满足 $\#f_{j_0}^{\text{mod}} = \alpha$.

第 6 步 输出 h .

定理 4.3 算法 3 正确. 算法需要 $O(nT \log D)$ 个单变元多项式插值, 这些多项式的次数小于 $O \sim (nDT)$, 项数小于 T . 除此之外, 算法还需要 $O \sim (nT^2 \log D)$ 次 $\mathcal{R}$ 中的环操作和 $O \sim (n^2 T^2 \log^2 D)$ 次位操作.

证明 因为 N_1 是使得 $p_1 \cdots p_{N_1} \geq D^{n(T-1)}$ 成立的最小整数, 所以, $2^{N_1-1} \leq p_1 p_2 \cdots p_{N_1-1} < D^{n(T-1)}$. 从而, $N_1 \leq \max\{1, \lceil n(T-1) \log_2 D \rceil\}$. 同理, $N_2 \leq \max\{1, \lceil nT \log_2 D \rceil\}$ 和 $N_3 \leq 4\{\max(1, \lceil nT \log_2 D \rceil)\}$. 所以, 在第 2 步中, 我们总是有 $K \geq N$. 根据定理 4.2 和引理 2.2 可知, f 的单项式至少一半在第 5 步的 **b** 中计算得到的 $\text{TS}_{(D, p_{j_0}, D)}^f$ 中. 在第 5 步的 **c** 中, 由定理 4.1, 我们可以计算 f 至少一半单项式. 算法的正确性已证.

现在估计算法的计算复杂度. 首先考虑单变元多项式插值次数. 令 $t = \#f$. 在第 3 步中, 我们调用了 N 个单变元多项式插值; 在第 5 步的 **a** 中, 调用了至多 $n \log_2 t$ 个单变元多项式插值. 由于第 i 个素数的大小是 $O(i \log i)$, 所以, N_1, N_2 和 N_3 大小为 $O(nT \log_2 D)$. 因此, $\deg(f_{(D, p_i)}) \leq D(p_i - 1) = O(nDT)$, $\deg(f_{(D, p_i, k)}) \leq 2D(p_i - 1) = O(nDT)$. 定理第一部分结论已证.

现在分析单变元多项式插值之外的计算复杂度. 在第 1 步中, 根据文献 [26, 定理 18.10], 计算前 K 个素数的位复杂度是 $O(K)$. 因为 $K = O(nT \log D)$, 所以, 第 1 步的位复杂度是 $O(nT \log D)$. 在第 2 步中, 需要计算 $D^{n(T-1)}$, D^{nT} 和 $D^{4n(T-1)}$. 由于数据的长度是 $O(nT \log D)$, 并且计算它们的算术步数是 $O(\log^2(nT))$, 所以, 位复杂度是 $O(nT \log D)$. 由于 N_1, N_2 和 N_3 的大小是 $O(nT \log D)$, 计算 N_1, N_2 和 N_3 的位复杂度是 $O(n^2 T^2 \log^2 D)$. 在第 3 步中, 计算一个 f_j^{mod} 需要 $O(T \log(pD))$ 次位操作和 $O(T)$ 次的环操作. 由于需要计算 N 个多项式 f_j^{mod} , 该步骤的复杂度是 $O(nT^2 \log^2 D)$ 次的位操作和 $O(nT^2 \log D)$ 次的环操作. 在第 4 步中, 由于 $N = O(nT \log D)$, f_j^{mod} 的项数不超过 T , 我们需要 $O(nT^2 \log D)$ 次位操作. 现在考虑第 5 步. 首先分析一个循环的复杂度. 根据引理 2.3, 步骤 **b** 的复杂度是 $O(nT \log D)$ 次位操作和 $O(nT)$ 次环操作. 在步骤 **c** 中, 由于每个检查需要比较 $N_1 + N_2 - 1$ 个多项式并进行 $N_1 + N_2 - 1$ 次替换, 所以, 复杂度是 $O(\#\text{TS}_{(D, p_{j_0}, D)}^{f-h} n(N_1 + N_2) \log(pD) + \#\text{TS}_{(D, p_{j_0}, D)}^{f-h} (N_1 + N_2) \log T \log(pD))$ 次位操作和 $O(\#\text{TS}_{(D, p_{j_0}, D)}^{f-h} (N_1 + N_2))$ 次环操作. 由于 $\#\text{TS}_{(D, p_{j_0}, D)}^{f-h} \leq T$, 复杂度是 $O(n^2 T^2 \log^2 D)$ 次位操作和 $O(nT^2 \log D)$ 次环操作. 与第 2 步类似, 步骤 **d** 的复杂度是 $O(n^2 T^2 \log^2 D)$ 次位操作. 在步骤 **e** 中, 需要 $n \cdot \#s$ 次操作去计算 $s_{(D, p_j)}$ 和 $s_{(D, p_j)}^{\text{mod}}$. 由于需要更新 N 个多项式, 其复杂度是 $O(n(\#s)N \log(pD))$ 次位操作和 $O(\#sN)$ 次环操作. 因此, 整个循环的复杂度在步骤 **c** 中达到. 经过每个循环, 得到 f 至少一半的多项式, 所以至多循环 $\log_2 t \leq O(\log T)$ 次循环. 所以, 第 5 步的复杂度是 $O(n^2 T^2 \log^2 D)$ 次位操作和 $O(nT^2 \log D)$ 次环操作, 这也是整个算法的复杂度. $\square$

4.3 多元多项式插值

类似于第 3.4 小节, 将算法 3 与三种不同的单变元插值算法组合, 我们可给出三个多元插值算法.

与推论 3.1 类似, 我们有以下结果.

推论 4.1 令 $\mathcal{R}$ 是一个包含至少 $O(nTD)$ 个元素的整环. 如果使用 Lagrange 算法计算单变元插值, 那么算法 3 需要 $O(n^2 T^2 D)$ 次对 $\mathcal{B}_f$ 的查询和 $O(n^2 T^2 D)$ 次 $\mathcal{R}$ 操作, 以及少量的位操作.

与推论 3.2 类似, 我们有以下结果.

推论 4.2 令 $\mathcal{R} = \mathbb{C}$. 若使用 Ben-Or & Tiwari 算法计算单变元插值, 则算法 3 需要 $O(nT^2 \log D)$ 次对 $\mathcal{B}_f$ 的查询和 $O(n^2 T^4 D)$ 次位操作.

与推论 3.3 类似, 我们有以下结果.

推论 4.3 令 $\mathcal{R} = \mathbb{F}_q$. 如果使用修改的 Ben-Or & Tiwari 算法计算单变元插值, 那么算法 3 需要 $O(nT^2 \log D)$ 次对 $\mathcal{B}_f$ 的查询和 $O(n^2 T^2 D \log q)$ 次位操作.

5 结论

本文通过引入新 Kronecker 类型的变元替换来将黑盒多元多项式插值约化成单变元多项式插值, 进而给出了新的多元多项式稀疏插值算法. 在域 $\mathbb{C}$ 中, 算法的位复杂度关于 D 是线性的, 而所有现存的其他确定性算法都是关于 D 二次的. 在有限域上, 新算法与现有的确定算法相比, 关于 T 和 D , 具有更好的复杂度; 关于 n 有相同的复杂度.

最后, 我们比较算法 2 和 3 在有限域情形下的复杂度. 对于其他域, 结论是相同的. 算法 2 和 3 的复杂性分别是 $O^{\sim}(nTD \max(nT, D) \log q)$ 和 $O^{\sim}(n^2 T^2 D \log q)$. 如果 $nT > D$, 则这两个算法具有相同的复杂度. 如果 $nT < D$, 则算法 2 和 3 的复杂度的比率是 $\frac{D}{nT}$, 也就是说, 对于大的 D , 算法 3 性能更好.

参考文献

- 1 Ben-Or M, Tiwari P. A deterministic algorithm for sparse multivariate polynomial interpolation. In: Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing. New York: ACM Press, 1988, 301–309
- 2 Giesbrecht M, Labahn G, Lee W. Symbolic-numeric sparse interpolation of multivariate polynomials. J Symbolic Comput, 2009, 44: 943–959
- 3 Kaltofen E L, Lakshman Y N. Improved sparse multivariate polynomial interpolation algorithms. In: Proceedings of the International Symposium on Symbolic and Algebraic Computation. Lecture Notes in Computer Science, vol. 358. Berlin: Springer, 1988, 467–474
- 4 Kaltofen E L, Lee W S. Early termination in sparse interpolation algorithms. J Symbolic Comput, 2003, 36: 365–400
- 5 Kaltofen E L, Lee W S, Yang Z. Fast estimates of Hankel matrix condition numbers and numeric sparse interpolation. In: Proceedings of the 2011 International Workshop on Symbolic-Numeric Computation. New York: ACM Press, 2011, 130–136
- 6 Arnold A, Roche D S. Multivariate sparse interpolation using randomized Kronecker substitutions. In: Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation. New York: ACM Press, 2014, 35–42
- 7 Garg S, Schost E. Interpolation of polynomials given by straight-line programs. Theoret Comput Sci, 2009, 410: 2659–2662
- 8 Giesbrecht M, Roche D S. Diversification improves interpolation. In: Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation. New York: ACM Press, 2011, 123–130
- 9 Huang Q L, Gao X S. Faster interpolation algorithms for sparse multivariate polynomials given by straight-line programs. J Symbolic Comput, 2020, 101: 367–386
- 10 Huang Q L, Gao X S. Revisit sparse polynomial interpolation based on randomized Kronecker substitution. In: Computer Algebra in Scientific Computing. Lecture Notes in Computer Science, vol. 11661. Cham: Springer, 2019, 215–235
- 11 Huang Q L, Gao X S. Sparse polynomial interpolation with finitely many values for the coefficients. In: Computer Algebra in Scientific Computing. Lecture Notes in Computer Science, vol. 10490. Cham: Springer, 2017, 196–209
- 12 Huang Q L. Sparse polynomial interpolation over fields with large or zero characteristic. In: Proceedings of the International Symposium on Symbolic and Algebraic Computation. New York: ACM Press, 2019, 219–226
- 13 Klivans A R, Spielman D. Randomness efficient identity testing of multivariate polynomials. In: Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing. New York: ACM Press, 2001, 216–223
- 14 Zippel R. Interpolating polynomials from their values. J Symbolic Comput, 1990, 9: 375–403
- 15 Arnold A, Giesbrecht M, Roche D S. Faster sparse multivariate polynomial interpolation of straight-line programs. J Symbolic Comput, 2016, 75: 4–24
- 16 Arnold A, Giesbrecht M, Roche D S. Faster sparse interpolation of straight-line programs. In: Computer Algebra in Scientific Computing. Lecture Notes in Computer Science, vol. 8136. Cham: Springer, 2013, 61–74
- 17 Avendaño M, Krick T, Pacetti A. Newton-Hensel interpolation lifting. Found Comput Math, 2006, 6: 82–120
- 18 Grigoriev D Y, Karpinski M, Singer M F. Fast parallel algorithms for sparse multivariate polynomial interpolation over finite fields. SIAM J Comput, 1990, 19: 1059–1063
- 19 Alon N, Mansour Y. ϵ -discrepancy sets and their application for interpolation of sparse polynomials. Inform Process Lett, 1995, 54: 337–342

- 20 Cuyt A, Lee W S. A new algorithm for sparse interpolation of multivariate polynomials. *Theoret Comput Sci*, 2008, 409: 180–185
- 21 Mansour Y. Randomized interpolation and approximation of sparse polynomials. *SIAM J Comput*, 1995, 24: 357–368
- 22 Giesbrecht M, Roche D S. Interpolation of shifted-lacunary polynomials. *Comput Complexity*, 2010, 19: 333–354
- 23 Bläser M, Jindal G. A new deterministic algorithm for sparse multivariate polynomial interpolation. In: *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*. New York: ACM Press, 2014, 51–58
- 24 Huang M D A, Rao A J. Interpolation of sparse multivariate polynomials over large finite fields with applications. *J Algorithms*, 1999, 33: 204–228
- 25 Javadi S M M, Monagan M. Parallel sparse polynomial interpolation over finite fields. In: *Proceedings of the 4th International Workshop on Parallel and Symbolic Computation*, vol. 10. Grenoble: ACM Press, 2010, 160–168
- 26 von zur Gathen J, Gerhard J. *Modern Computer Algebra*. Cambridge: Cambridge University Press, 1999
- 27 Huang Q L. An improved early termination sparse interpolation algorithm for multivariate polynomials. *J Syst Sci Complex*, 2018, 31: 539–551

Deterministic sparse interpolation of black-box multivariate polynomials using Kronecker type substitutions

Qiaolong Huang & Xiaoshan Gao

Abstract In this paper, we propose two new deterministic interpolation algorithms for a sparse multivariate polynomial given as a standard black-box by introducing new Kronecker type substitutions. Let $f \in \mathcal{R}[x_1, \dots, x_n]$ be a sparse black-box polynomial with a degree bound D . When $\mathcal{R}$ equals $\mathbb{C}$ or a finite field, our algorithms either have better bit complexity or better bit complexity in D than existing deterministic algorithms. In particular, in the case of deterministic algorithms for standard black-box models, our second algorithm has the current best complexity in D which is the dominant factor in the complexity.

Keywords sparse polynomial interpolation, Kronecker substitution, black-box model

MSC(2010) 41A05, 68W05, 68W30

doi: 10.1360/N012019-00018