Robust な疎多項式補間
Robust algorithms for sparse interpolation of multivariate polynomials

近藤 和希 *
Kazuki Kondo
東京理科大学大学院
Graduate School, Tokyo University of Science

関川 浩 †
Hiroshi Sekigawa
東京理科大学
Tokyo University of Science

Abstract

We propose robust algorithms for sparse interpolation of multivariate black-box polynomials based on a modified Ben-Or/Tiwari algorithm that uses roots of unity as inputs.

1 はじめに

項数 t の多変数多項式の black-box が与えられたとき，この black-box にいくつかの任意の点を入力して値を求め，その値をもとに多変数多項式を決定することを sparse interpolation（疎多項式補間）という．本稿では，入力を素数とする Ben-Or/Tiwari のアルゴリズムを基にして，入力を 1 のべき根に変えたときに，変数の数が多い，または指数部の値が非常に大きい場合におけるアルゴリズムの robust 性をさらに向上させる手法について述べる．以下，2 節において，先行研究である Ben-Or/Tiwari のアルゴリズムと，その派生手法である変形 Ben-Or/Tiwari のアルゴリズム，沼畑によるアルゴリズム及びその改良について述べる．3 節において，沼畑によるアルゴリズムの改良版に残る問題点とそれを解決する手法について述べる．その後，既存方法と提案手法を比較する実験を行う．最後に 4 節において，まとめを行う．

2 先行研究

Sparse な多変数多項式の補間を行うアルゴリズムとして，代入する値に素数を用いる Ben-Or/Tiwari のアルゴリズム [1]，及び代入する値に 1 のべき根を用いた派生手法である変形 Ben-Or/Tiwari のアルゴリズム [2] が知られている．

アルゴリズム 1 (変形 Ben-Or/Tiwari のアルゴリズム)
Input :

- f : 多変数多項式の black-box
- t : f の項数

*1418503@ed.tus.ac.jp
†sekigawa@rs.tus.ac.jp

- n: f の変数の数
- $D_1, \ldots, D_n$: f のそれぞれの変数の次数上界

Output: $f = \sum_{j=1}^{t} c_j x_1^{d_{j1}} \ldots x_n^{d_{jn}}$ となる係数 $c_1, \ldots, c_t$ と指数 $d_{11}, \ldots, d_{tn}$

注意 1
素数を代入する Ben-Or/Tiwari のアルゴリズムは次数上界 $D_1, \ldots, D_n$ の入力を必要としない．

Step 1
互いに素で $p_1 > D_1, \ldots, p_n > D_n$ であるような $p_1, \ldots, p_n \in \mathbb{Z}$ を選び，$2t$ 個の point における値

$$\alpha_s = f(\omega_1^s, \ldots, \omega_n^s) \qquad (0 \leq s \leq 2t-1)$$

を求める (ただし、$\omega_k = \exp(2\pi i/p_k)$)．また，$m = p_1 \cdots p_n$ とする．

Step 2

$$A(z) = \prod_{j=1}^{t}(z - \omega^{d_j}) = \sum_{h=0}^{t} \lambda_h z^h$$

として (ただし，$\lambda_t = 1$，$\omega = \exp(2\pi i/m)$)，以下の Hankel 行列を係数行列とする方程式

$$\begin{bmatrix} \alpha_0 & \alpha_1 & \ldots & \alpha_{t-1} \\ \alpha_1 & \alpha_2 & \ldots & \alpha_t \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{t-1} & \alpha_t & \ldots & \alpha_{2t-2} \end{bmatrix} \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_{t-1} \end{bmatrix} = - \begin{bmatrix} \alpha_t \\ \alpha_{t+1} \\ \vdots \\ \alpha_{2t-1} \end{bmatrix}$$

を解き，$\lambda_0, \ldots, \lambda_{t-1}$ を求める．

Step 3
1 変数多項式 $A(z)$ の根 ω^{d_j} を求める．

Step 4
ω^{d_j} から，$round(\log_\omega \omega^{d_j}) = d_j$ を求め（主値をとる），

$$d_j = d_{j1} \cdot \frac{m}{p_1} + \cdots + d_{jn} \cdot \frac{m}{p_n}$$

として，それぞれの指数 $d_{11}, \ldots, d_{tn}$ を求める．

Step 5
$\beta_j(x_1, \ldots, x_n) = x_1^{d_{j1}} \cdots x_n^{d_{jn}}$ として，$\beta_j(\omega_1, \ldots, \omega_n) = \omega^{d_j}$ であるから，以下の Vandermonde 行列を係数行列とする方程式

$$\begin{bmatrix} 1 & 1 & \ldots & 1 \\ \omega^{d_1} & \omega^{d_2} & \ldots & \omega^{d_t} \\ \vdots & \vdots & \ddots & \vdots \\ (\omega^{d_1})^{t-1} & (\omega^{d_2})^{t-1} & \ldots & (\omega^{d_t})^{t-1} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_t \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{t-1} \end{bmatrix}$$

を解き，係数 $c_1, \ldots, c_t$ を求める．

Step 6
係数 $c_1, \ldots, c_t$ と，指数 $d_{11}, \ldots, d_{tn}$ を返す．

上記のアルゴリズムを基にして，指数部に中国剰余定理を用いることで robust 性を向上させたアルゴリズムを沼畑が提案している [3]．

アルゴリズム 2 (沼畑によるアルゴリズム)
Input :

- f : 多変数多項式の black-box
- t : f の項数
- n : f の変数の数
- $D_1, \ldots, D_n$: f のそれぞれの変数の次数上界

Output : $f = \sum_{j=1}^{t} c_j x_1^{d_{j1}} \ldots x_n^{d_{jn}}$ となる係数 $c_1, \ldots, c_t$ と指数 $d_{11}, \ldots, d_{tn}$

Step 1
$\gcd(\tilde{D}_{1j}, \ldots, \tilde{D}_{lj}) = 1$ $(1 \leq j \leq n)$ となるような互いに素な自然数の組 $(\tilde{D}_{11}, \ldots, \tilde{D}_{1n}), \ldots, (\tilde{D}_{l1}, \ldots, \tilde{D}_{ln})$ を $D_k < \tilde{D}_{1k} \cdots \tilde{D}_{lk}$ を満たすように取る．

Step 2
$(\tilde{D}_{j1}, \ldots, \tilde{D}_{jn})$ を次数上界の入力として $(1 \leq j \leq l)$，変形 Ben-Or/Tiwari のアルゴリズムを適用する．

Step 3
Step 2 で得られる，それぞれの変数の指数部を $(\tilde{D}_{j1}, \ldots, \tilde{D}_{jn})$ で割った剰余から，中国剰余定理を用いて f の指数部を求める．このとき，Step 2 で得られた多項式の係数が等しい項を f における同一の項とみなす．

沼畑によるアルゴリズムでは，次数上界 D_k の代わりにより小さい $\tilde{D}_{jk}$ を用いて変形 Ben-Or/Tiwari のアルゴリズムを実行するため，誤差の影響を受けやすい m の値が小さくなり robust 性が向上する．ただし，係数が等しい項を f における同一の項とみなして中国剰余定理を適用し指数を求めるため，f に同一の係数を持つ項が複数存在すると失敗する．

沼畑によるアルゴリズムにおいて，ある $\tilde{D}_k$ が正しい次数上界であったときには，その変数に対しては正しい指数部が求められる．そこで正しい指数部が得られたとき，以降の Step においてその変数に 1 を代入して計算を簡単にすることで，沼畑によるアルゴリズムを改良することができる．

アルゴリズム 3 (沼畑によるアルゴリズムの改良)
Input :

- f : 多変数多項式の black-box
- t : f の項数
- n : f の変数の数
- $D_1, \ldots, D_n$: f のそれぞれの変数の次数上界

Output : $f = \sum_{j=1}^{t} c_j x_1^{d_{j1}} \ldots x_n^{d_{jn}}$ となる係数 $c_1, \ldots, c_t$ と指数 $d_{11}, \ldots, d_{tn}$

Step 1
$k = 1$ とする．

Step 2
$\gcd(\tilde{D}_{1j}, \ldots, \tilde{D}_{kj}) = 1$ $(1 \leq j \leq n')$ となるような互いに素な自然数の組 $(\tilde{D}_{k1}, \ldots, \tilde{D}_{kn'})$ を次数上界の入力として，変形 Ben-Or/Tiwari のアルゴリズムを適用する．

Step 3

$$D_j \leq \prod_k \tilde{D}_{kj}$$

を満たす D_j が存在するとき，その変数における指数部を中国剰余定理を用いて求める．そして，以降の計算においてはその変数に 1 を代入して計算する．

Step 4
すべての変数の指数が求まればアルゴリズムは完了する，そうでなければ $k = k + 1$ として *Step 2* へ戻る．

注意 2
アルゴリズムの Step 2 において，n' はその反復内での変数の数を表している．

沼畑によるアルゴリズムの改良版は変数の数を減らして計算を行うことができるので，元の沼畑によるアルゴリズムよりも計算時間を短縮させ，さらに robust 性を向上させることができる．ただし，元の沼畑によるアルゴリズムと同様に，f に同一の係数を持つ項が複数存在するとこのアルゴリズムは失敗する．

3　主結果

沼畑によるアルゴリズムの改良版には，以下のような問題点がある．

- 変数の数が少なくなり過ぎると，項の衝突が起きる場合がある．
- 最初の反復内ではすべての変数が残っている状態でアルゴリズムを適用しなければならない．

項の衝突とは，区別されていた 2 つの項がなんらかの原因により，区別できなくなってしまうことである．以下に例を挙げる．

例 1
$f = x_1^5 x_2^{20} + 2x_1^7 x_2^{20}$ とする．x_1 の指数が求まり，$x_1 = 1$ とすると，項の衝突が起きる．

例 2
$f = x_1^5 x_2^7 + 2x_1^{16} x_2^{20}$ とする．沼畑によるアルゴリズムの改良版の Step 2 において $(\tilde{D}_1, \tilde{D}_2) = (11, 13)$ とすると，その結果は $x_1^5 x_2^7 + 2x_1^5 x_2^7$ となり，項の衝突が起きる．

以上の 2 つの問題を解決する方法として次の手法を提案する．

3.1 項の衝突を避ける次数上界を選ぶ手法

沼畑によるアルゴリズムの改良版の Step 2 において，$\tilde{D}_k$ として項の衝突が起きにくい値を選ぶ．そのときの指標として，次の項の衝突確率を考える．

定義 1 (項の衝突確率)
t を項数，$D_1, \ldots, D_n$ を次数上界とし，$m = D_1 \cdots D_n$ とする．このとき，項の衝突が起きる確率 p は次のように表される．

$$p = 1 - \frac{m}{m} \cdot \frac{m-1}{m} \cdot \cdots \cdot \frac{m-t+1}{m}$$

注意 3
この p は，各項においてそれぞれの変数が次数上界未満のランダムな指数を持つと仮定して計算している．

項数 t を変化させていったとき，衝突確率が 1% 未満となるような $\tilde{D}_1, \ldots, \tilde{D}_n$ は表 1 のようになる．

表 1: 項の衝突を避ける $\tilde{D}_k$

t	m	$n=1$	$n=2$	$n=3$
10	4481	(4481)	(67, 71)	(17, 19, 23)
20	18912	(18913)	(137, 139)	(23, 29, 31)
30	43292	(43313)	(211, 223)	(31, 37, 41)
40	77623	(77641)	(277, 281)	(41, 43, 47)
50	121903	(121909)	(349, 353)	(47, 53, 59)

表 1 から，項数によって適切な $\tilde{D}_1, \ldots, \tilde{D}_n$ を選び，項の衝突確率を 1% 未満にすることができるので，沼畑によるアルゴリズムの改良版において，変数を減らすことによる項の衝突が起きる問題を解決することができる．

3.2 変数の数を減らす手法

変形 Ben-Or/Tiwarri のアルゴリズムにおいて，ある変数における black-box へ入力する値をすべて 1 にすることで，変数の数を減らして計算することができる．ただし，変数の数を減らすことで項の衝突が起きる場合があるので，前節と同様に項の衝突確率の考えを導入し，変数の数の指標を決める．

表 2: 項の衝突を避ける変数の数

t	m	$n=1$	$n=2$	$n=3$
10	4481	4481	68	17
20	18912	18912	139	28
30	43292	43292	209	36
40	77623	77623	280	44
50	121903	121903	350	51

表 2 の各 n に対する値は m の n 乗根であり，各変数における次数上界がその値より大きくなっていれば，項の衝突確率が 1% 未満となる．よって，変数の数が 3 以上であれば，この例ではそれぞれの変数の次数上界は 100 未満程度でよく，項の衝突が起こりにくいと考えられる．以上から，次のアルゴリズムを提案する．

アルゴリズム 4 (変数の数を減らす手法)
Input :

- f : 多変数多項式の black-box
- t : f の項数
- n : f の変数の数
- $D_1, \ldots, D_n$: f のそれぞれの変数の次数上界

Output : $f = \sum_{j=1}^{t} c_j x_1^{d_{j1}} \ldots x_n^{d_{jn}}$ となる係数 $c_1, \ldots, c_t$ と指数 $d_{11}, \ldots, d_{tn}$

Step 1
入力された次数上界 $(D_1, \ldots, D_n)$ を $(D_1, D_2, D_3), \ldots, (D_{n-2}, D_{n-1}, D_n)$ と分割する．

注意 4
変数の数が 3 で割り切れない場合には，例えば

$$(10, 20, 30, 40) \Rightarrow (10, 20, 30), (20, 30, 40)$$

のように分割する．

Step 2
Step 1 で分割したそれぞれの次数上界に対して変形 Ben-Or/Tiwari のアルゴリズムを計 $\lceil n/3 \rceil$ 回適用する．このとき，入力する次数上界に対応しない変数には 1 を代入して計算する．

Step 3
Step 2 の結果から f を構成する．

3.3 オーバーサンプリング手法

変形 Ben-Or/Tiwari のアルゴリズムにおいて，Step 2 に現れる Hankel 行列と Step 5 に現れる Vandermonde 行列は悪条件な行列である．従って，これらを係数行列とする方程式を解く代わりに，black-box への入力点を増やすことで行列のサイズを拡大し，最小二乗法によってより精度の良い近似解を得ることができる．具体的には，変形 Ben-Or/Tiwari のアルゴリズムにおける Step 1 を次のように変更する．

Step 1
互いに素で $p_1 > D_1, \ldots, p_n > D_n$ であるような $p_1, \ldots, p_n \in \mathbb{Z}$ を選び，$2T$ 個の point における値

$$\alpha_s = f(\omega_1^s, \ldots, \omega_n^s) \qquad (0 \leq s \leq 2T - 1,\ T > t)$$

を求める (ただし、$\omega_k = \exp(2\pi i / p_k)$)．また，$m = p_1 \cdots p_n$ とする．

3.4 数値実験

今回提案した手法と既存のアルゴリズムを数値実験により比較する．実験環境として，OS は Windows 10 Pro，数式処理システムは Mathematica 11.3，CPU は Intel(R) Core(TM) i7-7500U CPU @ 2.70 GHz – 2.90 GHz，メモリは 16.0 GB のものを用いる．実験は計算精度を 20 とし，多変数多項式の black-box を係数が -10 から 10 までのランダムな実数, 指数が与えられた次数上界未満の非負整数として与える．このような設定の下で，100 個の多項式に対して補間を行い，その成功した個数と計算時間を調べる．

以下では，変形 Ben-Or/Tiwari のアルゴリズムを GLL，沼畑によるアルゴリズムを N，沼畑によるアルゴリズムの改良版を Modified N，項の衝突を避ける次数上界を選ぶ手法を Method 1，変数の数を減らす手法を Method 2 で表す．また，n は変数の数，D は次数上界を表す．実験結果は表 3 から表 6 である．

表 3: $n = 5$, $D = (10, 50, 100, 500, 1000)$

	no over sampling		$T = 2t$ over sampling	
Algorithms	time (sec)	success	time (sec)	success
GLL	18.1	11	141.0	73
N	37.2	8	274.5	74
Modified N	63.8	15	263.6	79
Method 1	103.8	27	372.8	93
Method 2	54.8	99	175.8	98

表 4: $n = 5$, $D = (100, 100, 100, 100, 100)$

	no over sampling		$T = 2t$ over sampling	
Algorithms	time (sec)	success	time (sec)	success
GLL	16.9	98	64.2	79
N	35.3	16	126.5	81
Modified N	35.5	16	125.3	81
Method 1	35.1	22	119.8	98
Method 2	24.1	100	98.6	96

表 5: $n = 7$, $D = (10, 50, 100, 500, 1000, 5000, 10000)$

	no over sampling		$T = 2t$ over sampling	
Algorithms	time (sec)	success	time (sec)	success
GLL	26.9	0	80.9	0
N	84.5	0	252.8	10
Modified N	70.0	0	218.0	28
Method 1	77.1	0	237.5	57
Method 2	45.5	83	180.0	82

表 6: $n = 10$, $D = (100, 200, 300, 400, 500, 600, 700, 800, 900, 1000)$

	no over sampling		$T = 2t$ over sampling	
Algorithms	time (sec)	success	time (sec)	success
GLL	37.6	0	101.4	0
N	78.5	0	207.7	0
Modified N	78.7	0	206.2	0
Method 1	79.8	0	211.5	2
Method 2	60.2	96	241.7	98

これらの結果から，今回提案した 2 つの手法の優位性が分かる．この 2 つの手法を比べると，Method 2 は常に変数の数を 3 として GLL を適用することができるので，Method 1 よりも robust であることが分かる．計算時間の観点からみると，Method 2 の計算時間は GLL のおおよそ 1.5 倍程度になっており，この点からも他のアルゴリズムよりも優位性があるといえる．また，オーバーサンプリング手法を用いるとより robust になることが実験結果から分かる．ただし，オーバーサンプリングをすることで行列のサイズが大きくなり，計算時間が大幅に増加している．

4 おわりに

今回，沼畑によるアルゴリズムの改良版の問題点とそれを解決する手法を提案した．今後の課題として，Hankel 行列の誤差を減らす方法と，より良い $\tilde{D}$ の取り方を検討していく．

謝辞

本研究は科研費 18K11172 の助成を受けたものである．

参 考 文 献

[1] M. Ben-Or and P. Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. In *Proc. Twentieth Annual ACM Symp. Theory Comput.*, 301–309, 1988.

[2] M. Giesbrecht, G. Labahn, and W.-s. Lee. Symbolic-numeric sparse interpolation of multivariate polynomials. *J. Symbolic Comput.*, 44:943–959, 2009.

[3] D. Numahata and H. Sekigawa. Robust algorithms for sparse interpolation of multivariate polynomials. *ACM Communications in Computer Algebra*, to appear.