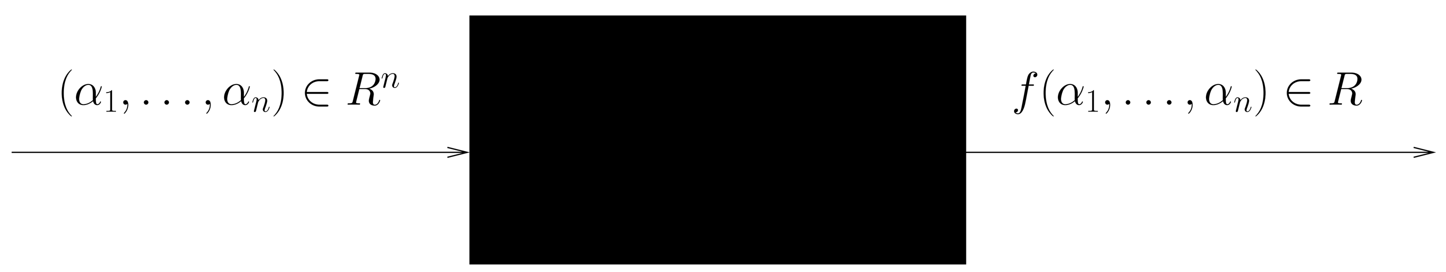
On Sparse Interpolation over Finite Fields

Seyed Mohammad Mahdi Javadi, Michael Monagan

sjavadi@cs.sfu.ca, mmonagan@cecm.sfu.ca

The Problem

The problem of interpolating multivariate polynomials over a finite field is one of the most challenging problems in *computer algebra*. It has been of interest for a long time and has many applications and many solutions.



Let f be a multivariate polynomial in variables $x_1, \ldots, x_n$ with t non-zero terms. The problem is given a black box that on input $\alpha_1, \ldots, \alpha_n$ outputs $f(x_1 = \alpha_1, \ldots, x_n = \alpha_n)$, we want to find the *target polynomial* $f(x_1 \ldots, x_n)$ by *probing* the black box at a series of evaluation points.

Newton's Interpolation Algorithm

The classical method is Newton's algorithm:

1. Let d be a bound on the degree of f in each variable x_i
2. Choose $\beta_1, \beta_2, \ldots, \beta_{d+1}$ random points
3. Recursively interpolate $f_i = f(x_1 = \beta_i, x_2, \ldots, x_n)$ for $1 \le i \le d+1$
4. Use the Chinese remaindering algorithm to interpolate f from $f_1, \ldots, f_{d+1}$

Newton's algorithm does $(d+1)^n$ probes to the black box.

Example 1. *For $f = x_1^d + x_2^d + \cdots + x_n^d + 1$, Newton's algorithm does $(d+1)^n$ probes even though f has only $n+1$ non-zero terms.*

Zippel's Sparse Interpolation Algorithm

The number of probes in Zippel's sparse interpolation algorithm is polynomial in t, the number of non-zero terms in the target polynomial f.

Idea: After interpolating the first image $f_1 = f(x_1 = \beta_1)$, one can use the form of f_1 to compute $f_2, \ldots, f_{d+1}$. This is done by solving systems of linear equations.

Example 2. *Let $f = 4x^{13}y^2 - 3x^5 + 4y^3 - 1$. Let $\beta_1 = 2$. We first interpolate $f_1 = f(y = \beta_1) = 16x^{13} - 3x^5 + 31$ using 14 probes to the black box. We assume the form for f: $g = Ax^{13} + Bx^5 + C$. Each f_i now can be computed using 3 probes to the black box.*

Zippel's algorithm does $O(ndt)$ probes to the black box.

Problem: The number of probes in Zippel's algorithm still depends on a bound d on the degree of f in each variable.

Ben-Or/Tiwari Sparse Interpolation Algorithm

Let f be a polynomial with coefficients in $\mathbb{Z}$. In Ben-Or/Tiwari sparse interpolation algorithm, the number of probes does not depend on the degree. It only depends on T, a bound on the number of non-zero terms in f.

1. Let $p_1, p_2, \ldots, p_n$ be the first n prime numbers.
2. For $i = 0, \ldots, 2T-1$, Let b_i be the output of black box on $(p_1^i, \ldots, p_n^i)$.
3. Find the λ_i s.t. $b_{t+i} = \lambda_{t-1} b_{t+i-1} + \lambda_{t-2} b_{t+i-2} + \cdots + \lambda_0 b_i$ for all $i \ge 0$.
4. Let $\Lambda(z) = z^t - \lambda_{t-1} z^{t-1} - \cdots - \lambda_0$.
5. Compute $r_1, \ldots, r_t$, the integer roots of $\Lambda(z)$.
6. Each r_i is equal to a monomial of f evaluated at $(x_1 = p_1, x_2 = p_2, \ldots, x_n = p_n)$. Find the monomials using integer divisions.
7. Find the coefficients of f by solving a system of linear equations.

Ben-Or/Tiwari algorithm does $2T$ probes to the black box.

Example 3. *Let $f(x, y) = 4x^{13}y^2 - 3x^5 + 4y^3 - 1$. We have $p_1 = 2, p_2 = 3$. Let $T = 4$ be the bound on the number of terms in f. We have*

$$b_0 = f(p_1^0, p_2^0) = 4, b_1 = f(p_1^1, p_2^1) = 294923,$$

$$b_2 = f(p_1^2, p_2^2) = 21743271779, b_3 = f(p_1^3, p_2^3) = 1603087953277835,$$

$$b_4 = f(p_1^4, p_2^4) = 118192468620710277059, b_5 = f(p_1^5, p_2^5) = 8714094326467802463717803,$$

$$b_6 = f(p_1^6, p_2^6) = 642472746501818143233353336099,$$

$$b_7 = f(p_1^7, p_2^7) = 47368230654086048064431853086526155.$$

Using the **Berlekamp/Massey** *algorithm we find the linear generator for this sequence:*

$$\Lambda(z) = z^4 - 73788\, z^3 + 4424603\, z^2 - 68051808\, z + 63700992.$$

The roots of this polynomial are $73728 = p_1^{13} \times p_2^2$, $32 = p_1^5$, $27 = p_2^3$ and 1. Hence the monomials are $x^{13}y^2, x^5, y^3$ and 1.

Problem: Unfortunately one can not use this algorithm for a polynomial over a finite field unless the characteristic p is very large. Let $f = \sum_{i=1}^{t} C_i M_i \in \mathbb{Z}_p[x_1, \ldots, x_n]$. Choose $(\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}_p^n$ at random. One can use Steps 1 to 5 of the Ben-Or/Tiwari algorithm to find the images of the monomials $r_i = M_i(\alpha_1, \ldots, \alpha_n) \bmod p$. The problem is that we can not uniquely determine the degrees of the monomials by their images $r_1, \ldots, r_t$ using only integer divisions in $\mathbb{Z}_p$.

Our New Sparse Interpolation Algorithm

Our sparse interpolation algorithm is a modification of the Ben-Or/Tiwari algorithm for polynomials over finite fields. It costs an extra factor of $O(n)$ probes.

Idea: We choose the evaluation point $(\alpha_1, \ldots, \alpha_n, \alpha_{n+1}) \in \mathbb{Z}_p^{n+1}$ at random. We first run the first five steps of the Ben-Or/Tiwari algorithm to find the images of the monomials $r_i = M_i(\alpha_1, \ldots, \alpha_n)$. To find the degrees of the monomials in the variable x_j, we replace α_j by α_{n+1}. We run the first 5 steps again and we find $\bar{r}_i = M_i(\alpha_1, \ldots, \alpha_{j-1}, \alpha_{n+1}, \alpha_{j+1}, \ldots, \alpha_n)$.

Observation: We have

$$\frac{r_i}{\bar{r}_i} = \left(\frac{\alpha_j}{\alpha_{n+1}}\right)^{d_i},$$

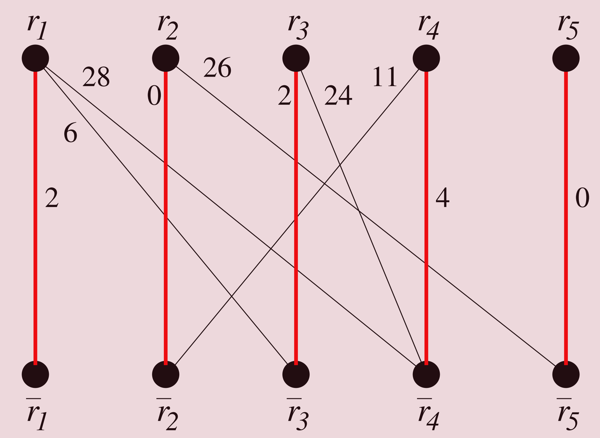
where $d_i = \deg_{x_j}(M_i)$. We will use this fact to find the degrees of all the monomials in x_j. The problem is we need to match the root r_i with the corresponding root $\bar{r}_i$. To do this, we use bipartite matching algorithm from graph theory.

Our new algorithm does $2nT$ probes to the black box.

Example 4. *Let $f = 25y^2z + 90yz^2 + 93x^2y^2z + 60y^4z + 42z^5 \in \mathbb{Z}_{101}[x, y, z]$. Here $t = 5, n = 3$. Suppose we now that the degree bound on the degree of f in each variable is $d = 40$. We choose the following evaluation points $\alpha_1 = 85, \alpha_2 = 96, \alpha_3 = 58$ and $\alpha_4 = 99$. Suppose we want to find the degrees of the monomials in y. We run the first steps of the Ben-Or/Tiwari algorithm for both $\beta_1 = (x = \alpha_1, y = \alpha_2, z = \alpha_3)$ and $\beta_2 = (x = \alpha_1, y = \alpha_4, z = \alpha_3)$. We obtain two sets of roots $R = \{36, 47, 25, 92, 87\}$ and $\bar{R} = \{30, 39, 4, 19, 87\}$. Let the graph G be a bipartite graph with nodes R and $\bar{R}$ such that r_i is connected to $\bar{r}_j$ if and only if*

$$\frac{r_i}{\bar{r}_j} = \left(\frac{\alpha_2}{\alpha_4}\right)^e,$$

for some $0 \le e \le d = 40$. We have



We try to find a perfect matching in this graph. The edges which are in the perfect matching are highlighted in red. We find that the degrees of the monomials in y are $2, 1, 2, 4$ and 0.

Protobox

In 2000, Kaltofen *et al.*, presented a hybrid of Zippel and Ben-Or Tiwari algorithms which they call a racing algorithm. To interpolate the next variable, their algorithm runs a Newton interpolation and univariate Ben-Or/Tiwari algorithm, stopping when the first succeeds to reduce the number of probes. The purpose of the early termination technique is to avoid using bounds for determining the termination point in an algorithm. Instead the racing algorithm stops when the interpolated polynomial does not change after a certain number of probes to the black box.

Benchmarks

$f_i \in \mathbb{Z}_p[x_1, \ldots, x_6]$ where $p = 3037000453$. We have $\#f_i \approx 2^i$ and $d = 30$. DNF means "Did Not Finish".

i	$\#f$	New Algorithm		Zippel		ProtoBox
		Time	Probes	Time	Probes	Probes
1	2	0.00	24	0.01	496	37
2	3	0.00	36	0.01	651	59
3	8	0.00	96	0.01	1364	140
4	16	0.00	192	0.02	2511	284
5	31	0.00	372	0.05	4340	521
6	64	0.02	768	0.15	8060	995
7	127	0.06	1524	0.44	14601	1871
8	255	0.21	3060	1.51	27652	3615
9	511	0.81	6132	5.19	50530	6692
10	1016	3.10	12192	17.94	90985	12591
11	2037	12.20	24444	65.35	168299	DNF
12	4083	48.06	48996	230.60	301320	DNF
13	8151	189.21	97812	803.26	532549	DNF

References

[1] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *EUROSAM '79: Proc. of the International Symposiumon on Symbolic and Algebraic Computation*, pages 216–226, London, UK, 1979.

[2] P. Tiwari M. Ben-Or. A deterministic algorithm for sparse multivariate polynomial interpolation. In *STOC '88: Proc. of the twentieth annual ACM symposium on Theory of computing*, pages 301–309, 1988. ACM.

[3] Erich Kaltofen and Wen-shin Lee. Early termination in sparse interpolation algorithms. *J. Symb. Comput.*, 36(3-4):365–400, 2003.