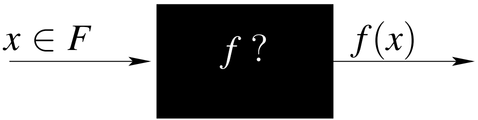
(Sparse) Interpolation with Outliers

Clément PERNET[†]
joint work with Matthew T. COMER[*] and Erich L. KALTOFEN[*]

[†]: LIG/INRIA-MOAIS, Grenoble Université, France
[*]: North Carolina State University, USA

SIAM Applied Algebraic Geometry 2013,
Fort Collins, CO, USA
Aug 3rd, 2013

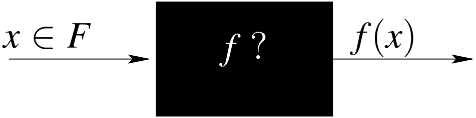
Motivation : model fitting



Problem

Recover an unknown function f, given as a black-box, from its evaluations.

Motivation : model fitting

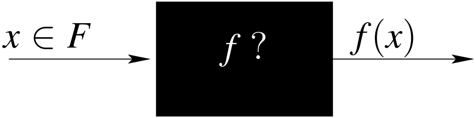


$$f = \sum_{i=0}^{k} c_i x^i$$

Additional knowledge on the shape f

Dense Polynomial: degree bound

Motivation : model fitting



$$f = \sum_{i=1}^{t} c_i x^{e_i}$$

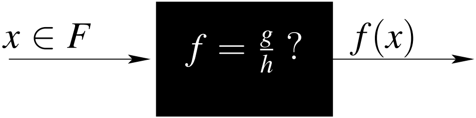
Additional knowledge on the shape f

Dense Polynomial: degree bound

Sparse polynomial:

- support: location of non zero terms
- sparsity: number of non zero terms

Motivation : model fitting



$$g = \sum_{i=0}^{d_G} g_i x^i, \quad h = \sum_{i=0}^{d_H} h_i x^i$$

Additional knowledge on the shape f

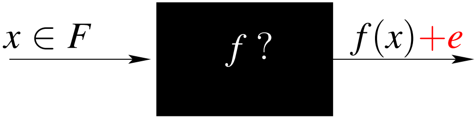
Dense Polynomial: degree bound

Sparse polynomial:

- support: location of non zero terms
- sparsity: number of non zero terms

Dense Rational function: degree bounds

Motivation : model fitting



Additional knowledge on the shape f

Dense Polynomial: degree bound

Sparse polynomial:

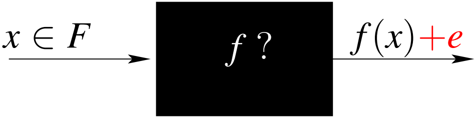
- support: location of non zero terms
- sparsity: number of non zero terms

Dense Rational function: degree bounds

Trust in the evaluations

- approximations: numerical noise
- true errors

Motivation : model fitting



Additional knowledge on the shape f

Dense Polynomial: degree bound

Sparse polynomial:

- support: location of non zero terms
- sparsity: number of non zero terms

Dense Rational function: degree bounds

Trust in the evaluations

- approximations: numerical noise
- true errors

Outline

Dense polynomial interpolation with errors

Sparse polynomial interpolation with errors

- de Proni/Ben-Or/Tiwari interpolation
- Fault tolerant Berlekamp/Massey algorithm
- Relations to Reed-Solomon decoding

Dense rational function interpolation with errors

Outline

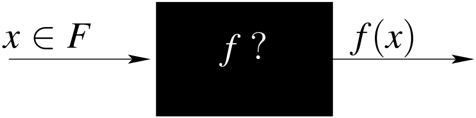
Dense polynomial interpolation with errors

Sparse polynomial interpolation with errors

- de Proni/Ben-Or/Tiwari interpolation
- Fault tolerant Berlekamp/Massey algorithm
- Relations to Reed-Solomon decoding

Dense rational function interpolation with errors

Dense polynomial recovery

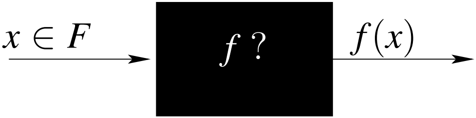


$$f = \sum_{i=0}^{k} c_i x^i$$

without error: polynomial interpolation (Lagrange, Newton, etc).

$$f(X) = \sum_{i=0}^{k} y_i \frac{L_i(X)}{L_i(x_i)}, \text{ with } L_i = \prod_{j \neq i} (X - x_j)$$

Dense polynomial recovery



$$f = \sum_{i=0}^{k} c_i x^i$$

without error: polynomial interpolation (Lagrange, Newton, etc).

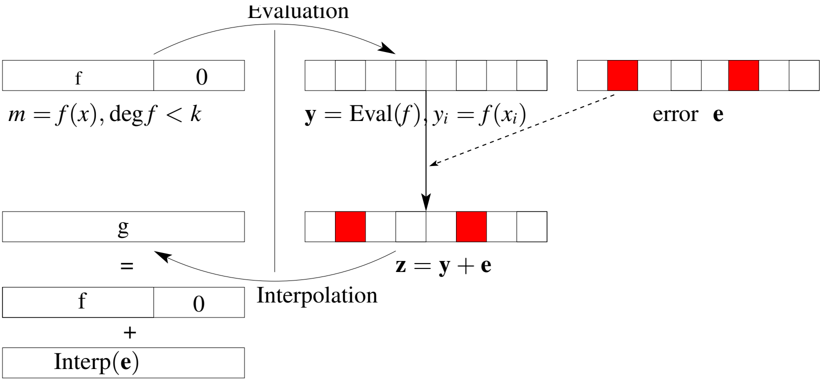
$$f(X) = \sum_{i=0}^{k} y_i \frac{L_i(X)}{L_i(x_i)}, \text{ with } L_i = \prod_{j \neq i} (X - x_j)$$

with errors: Reed-Solomon decoding

- $y_i = f(x_i) + e_i$ where the vector $\mathbf{e}$ is t-sparse.
- Interp$(\mathbf{y}) = f +$ Interp$(\mathbf{e})$
- [Blahut, 1984]: Interp$(\mathbf{e})$ has linear cpxty t
- Berlekamp-Massey: error locator from the linear generating relation

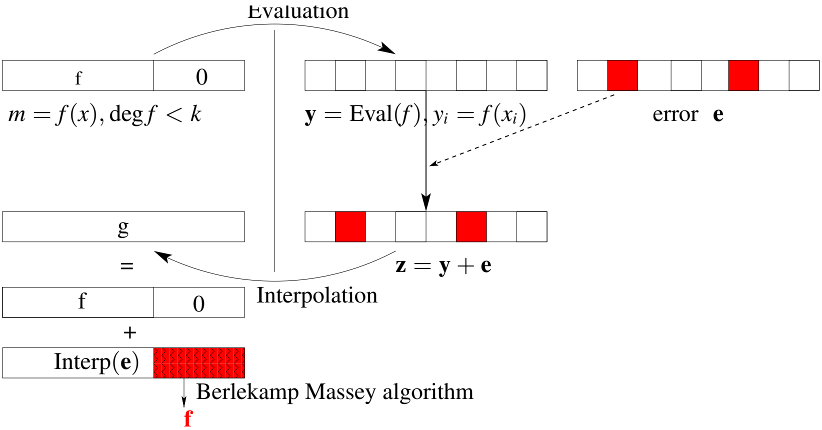
Reed-Solomon codes as Evaluation codes

$$\mathcal{C} = \{(f(x_1), \ldots, f(x_n)) \mid \deg f < k\}$$



Reed-Solomon codes as Evaluation codes

$$\mathcal{C} = \{(f(x_1), \ldots, f(x_n)) \mid \deg f < k\}$$



Parameter oblivious decoding

Improving the correction capacity:

- With a fixed number n of evaluations, the correction capacity depends on the degree of f:

  can correct up to $E \leq \frac{n - \deg f - 1}{2}$

  $\Rightarrow$ bounds on $\deg f$: often pessimistic

Parameter oblivious decoding

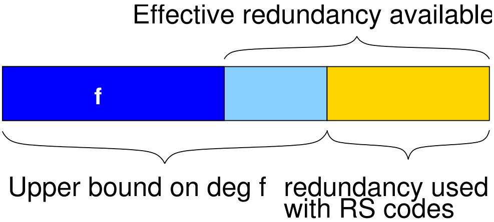
Improving the correction capacity:

- With a fixed number n of evaluations, the correction capacity depends on the degree of f:

  can correct up to $E \leq \frac{n - \deg f - 1}{2}$

  $\Rightarrow$bounds on $\deg f$: often pessimistic

  $\Rightarrow$how to take advantage of all the available redundancy?



Parameter oblivious decoding

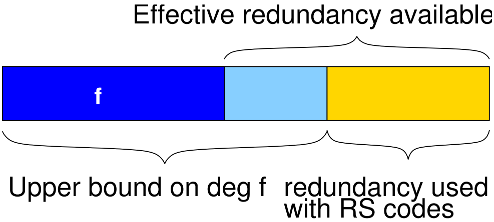
Improving the correction capacity:

- With a fixed number n of evaluations, the correction capacity depends on the degree of f:

  can correct up to $E \leq \frac{n - \deg f - 1}{2}$

  ⇒bounds on $\deg f$: often pessimistic

  ⇒how to take advantage of all the available redundancy?



- Achieved with Ext. Euclidean Alg. with various termination criteria [Khonji, Pernet, Roch, Roche and Stalinski, 2010]:
  - divisibility check
  - quotient likely to be large upon decoding iteration

Outline

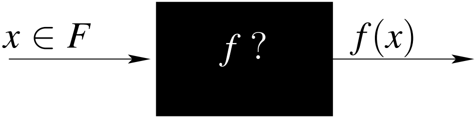
Dense polynomial interpolation with errors

Sparse polynomial interpolation with errors

- de Proni/Ben-Or/Tiwari interpolation
- Fault tolerant Berlekamp/Massey algorithm
- Relations to Reed-Solomon decoding

Dense rational function interpolation with errors

Sparse Polynomial Interpolation

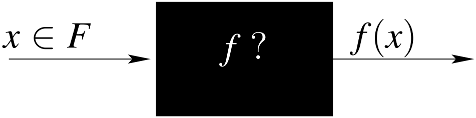


$$f = \sum_{i=1}^{t} c_i x^{e_i}$$

Problem

Recover a t-sparse polynomial f given a black-box computing evaluations of it.

Sparse Polynomial Interpolation



$$f = \sum_{i=1}^{t} c_i x^{e_i}$$

Problem

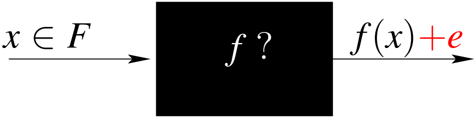
Recover a t-sparse polynomial f given a black-box computing evaluations of it.

[Ben-Or and Tiwari, 1988]

- Let $a_i = f(p^i)$ for p an element, and $\Lambda(z) = \prod_{i=1}^{t}(z - p^{e_i})$.
- Then $\Lambda(z)$ is the minimal generator of the seq. $(a_0, a_1, \ldots)$.

$\Rightarrow$only $2t$ entries needed to find $\Lambda(\lambda)$ (Berlekamp-Massey)

Sparse Polynomial Interpolation with errors



Problem

Recover a t-sparse polynomial f given a black-box computing evaluations of it.

[Ben-Or and Tiwari, 1988]

- Let $a_i = f(p^i)$ for p an element, and $\Lambda(z) = \prod_{i=1}^{t}(z - p^{e_i})$.
- Then $\Lambda(z)$ is the minimal generator of the seq. $(a_0, a_1, \dots)$.

$\Rightarrow$only $2t$ entries needed to find $\Lambda(\lambda)$ (Berlekamp-Massey)

[Comer, Kaltofen and Pernet, 2012]

$\Rightarrow$only $2t(2E+1)$ entries needed with $e \leq E$ errors.
using a fault-tolerant Berlekamp-Massey algorithm

Fault tolerant Berlekam/Massey algorithm

Problem statement

Suppose $(a_0, a_1, \dots)$ is linearly generated by $\Lambda(z)$ of degree t where $\Lambda(0) \neq 0$.
Given $(b_0, b_1, \dots) = (a_0, a_1, \dots) + \varepsilon$, where weight$(\varepsilon) \leq E$:

1. How to recover $\Lambda(z)$ and $(a_0, a_1, \dots)$
2. How many entries required for
   - a unique solution
   - a list of solutions containing $(a_0, a_1, \dots)$

Fault tolerant Berlekam/Massey algorithm

Problem statement

Suppose $(a_0, a_1, \dots)$ is linearly generated by $\Lambda(z)$ of degree t where $\Lambda(0) \neq 0$.
Given $(b_0, b_1, \dots) = (a_0, a_1, \dots) + \varepsilon$, where weight$(\varepsilon) \leq E$:

1. How to recover $\Lambda(z)$ and $(a_0, a_1, \dots)$
2. How many entries required for
   - a unique solution
   - a list of solutions containing $(a_0, a_1, \dots)$

Coding Theory formulation

Let $\mathcal{C}$ be the set of all sequences of linear complexity t.

1. How to decode $\mathcal{C}$?
2. What are the best correction capacity ?
   - for unique decoding
   - list decoding

How many entries to guarantee uniqueness?

Case $E = 1, t = 2$

$$\begin{array}{ccccccccccc|l} & & & & & (a_i) & & & & & & \Lambda(z) \\ (0, & 1, & 0, & 1, & 0, & 1, & 0, & -1, & 0, & 1, & 0) & 2 - 2z^2 + z^4 + z^6 \end{array}$$

Where is the error?

How many entries to guarantee uniqueness?

Case $E = 1, t = 2$

(a_i)	$\Lambda(z)$
$(0, 1, 0, 1, 0, 1, 0, -1, 0, 1, 0)$	$2 - 2z^2 + z^4 + z^6$
$(0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0)$	$-1 + z^2$

Where is the error?

How many entries to guarantee uniqueness?

Case $E = 1, t = 2$

$$\begin{array}{ccccccccccc|l} & & & & & (a_i) & & & & & & \Lambda(z) \\ (0, & 1, & 0, & 1, & 0, & 1, & 0, & -1, & 0, & 1, & 0) & 2 - 2z^2 + z^4 + z^6 \\ (0, & 1, & 0, & 1, & 0, & 1, & 0, & \color{red}{1}, & 0, & 1, & 0) & -1 + z^2 \\ (0, & 1, & 0, & \color{red}{-1}, & 0, & 1, & 0, & -1, & 0, & 1, & 0) & 1 + z^2 \end{array}$$

Where is the error?

How many entries to guarantee uniqueness?

Case $E = 1, t = 2$

$$
\begin{array}{ccccccccccc|l}
 & & & & & (a_i) & & & & & & \Lambda(z) \\
(0, & 1, & 0, & 1, & 0, & 1, & 0, & -1, & 0, & 1, & 0) & 2 - 2z^2 + z^4 + z^6 \\
(0, & 1, & 0, & 1, & 0, & 1, & 0, & \color{red}{1}, & 0, & 1, & 0) & -1 + z^2 \\
(0, & 1, & 0, & \color{red}{-1}, & 0, & 1, & 0, & -1, & 0, & 1, & 0) & 1 + z^2
\end{array}
$$

Where is the error?

A unique solution is not guaranteed with $t = 2, E = 1$ and $n = 11$

Is $n \geq 2t(2E + 1)$ a necessary condition?

Generalization to any $E \geq 1$

Let $\overline{0} = (\overbrace{0, \ldots, 0}^{t-1 \text{ times}})$. Then

$$s = (\overline{0}, 1, \overline{0}, 1, \overline{0}, 1, \overline{0}, -1)$$

is generated by $z^t - 1$ or $z^t + 1$ up to $E = 1$ error.
Then

$$(\overbrace{s, s, \ldots, s}^{E \text{ times}}, \overline{0}, 1, \overline{0})$$

is generated by $z^t - 1$ or $z^t + 1$ up to E errors.
$\Rightarrow$ambiguity with $n = 2t(2E + 1) - 1$ values.

Generalization to any $E \geq 1$

Let $\overline{0} = (\overbrace{0, \ldots, 0}^{t-1 \text{ times}})$. Then

$$s = (\overline{0}, 1, \overline{0}, 1, \overline{0}, 1, \overline{0}, -1)$$

is generated by $z^t - 1$ or $z^t + 1$ up to $E = 1$ error.
Then

$$(\overbrace{s, s, \ldots, s}^{E \text{ times}}, \overline{0}, 1, \overline{0})$$

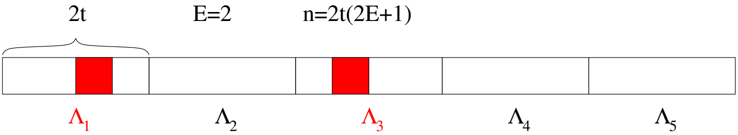
is generated by $z^t - 1$ or $z^t + 1$ up to E errors.
$\Rightarrow$ambiguity with $n = 2t(2E + 1) - 1$ values.

Theorem

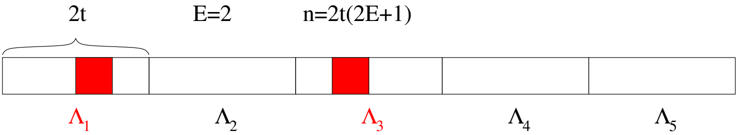
Necessary condition for unique decoding:

$$n \geq 2t(2E + 1)$$

The Majority Rule Berlekamp/Massey algorithm



The Majority Rule Berlekamp/Massey algorithm



Input: $(a_0, \ldots, a_{n-1}) + \varepsilon$, where $n = 2t(2E+1)$, $weight(\varepsilon) \le E$, and $(a_0, \ldots, a_{n-1})$ minimally generated by Λ of degree t, where $\Lambda(0) \ne 0$.

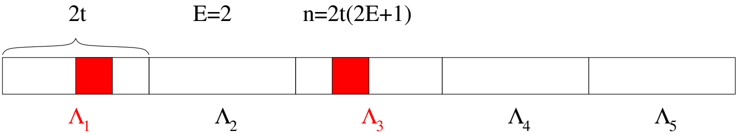
Output: $\Lambda(z)$ and $(a_0, \ldots, a_{n-1})$.

begin

Run BMA on $2E+1$ segments of $2t$ entries and record $\Lambda_i(z)$ on each segment;

Perform majority vote to find $\Lambda(z)$;

The Majority Rule Berlekamp/Massey algorithm



Input: $(a_0, \dots, a_{n-1}) + \varepsilon$, where $n = 2t(2E+1)$, $weight(\varepsilon) \leq E$, and $(a_0, \dots, a_{n-1})$ minimally generated by Λ of degree t, where $\Lambda(0) \neq 0$.

Output: $\Lambda(z)$ and $(a_0, \dots, a_{n-1})$.

begin

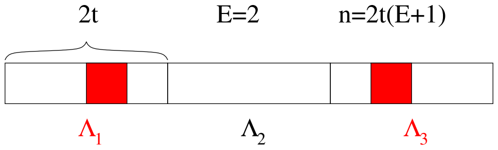
Run BMA on $2E+1$ segments of $2t$ entries and record $\Lambda_i(z)$ on each segment;

Perform majority vote to find $\Lambda(z)$;

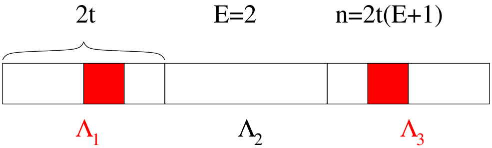
Use a *clean* segment to *clean-up* the sequence ;

return $\Lambda(z)$ and $(a_0, a_1, \dots)$;

List decoding for $n \geq 2t(E+1)$



List decoding for $n \geq 2t(E+1)$



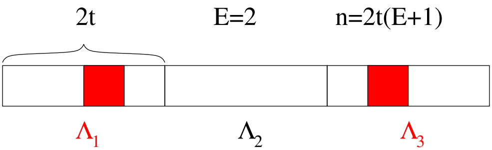
Input: $(a_0, \ldots, a_{n-1}) + \varepsilon$, where $n = 2t(E+1)$, $weight(\varepsilon) \leq E$, and $(a_0, \ldots, a_{n-1})$ minimally generated by Λ of degree t, where $\Lambda(0) \neq 0$.

Output: $(\Lambda_i(z), s_i = (a_0^{(i)}, \ldots, a_{n-1}^{(i)}))_i$ a list of $\leq E$ candidates

begin

Run BMA on $E+1$ segments of $2t$ entries and record $\Lambda_i(z)$ on each segment;

List decoding for $n \geq 2t(E+1)$



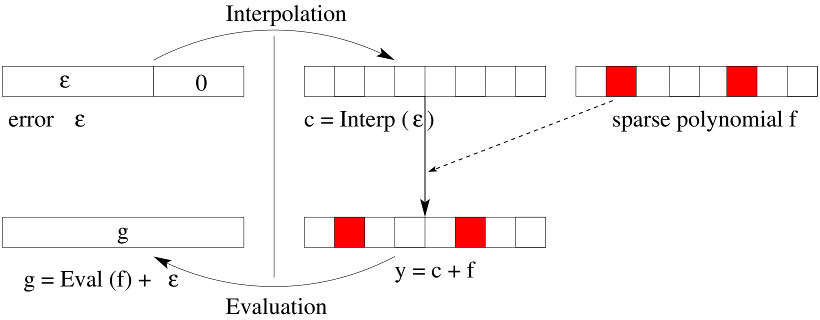
Input: $(a_0, \ldots, a_{n-1}) + \varepsilon$, where $n = 2t(E+1)$, $weight(\varepsilon) \leq E$, and $(a_0, \ldots, a_{n-1})$ minimally generated by Λ of degree t, where $\Lambda(0) \neq 0$.

Output: $(\Lambda_i(z), s_i = (a_0^{(i)}, \ldots, a_{n-1}^{(i)}))_i$ a list of $\leq E$ candidates

begin
- Run BMA on $E+1$ segments of $2t$ entries and record $\Lambda_i(z)$ on each segment;
- **foreach** $\Lambda_i(z)$ **do**
  - Use a *clean* segment to *clean-up* the sequence;
  - Withdraw Λ_i if no clean segment can be found.
- **return** the list $(\Lambda_i(z), (a_0^{(i)}, \ldots, a_{n-1}^{(i)}))_i$;

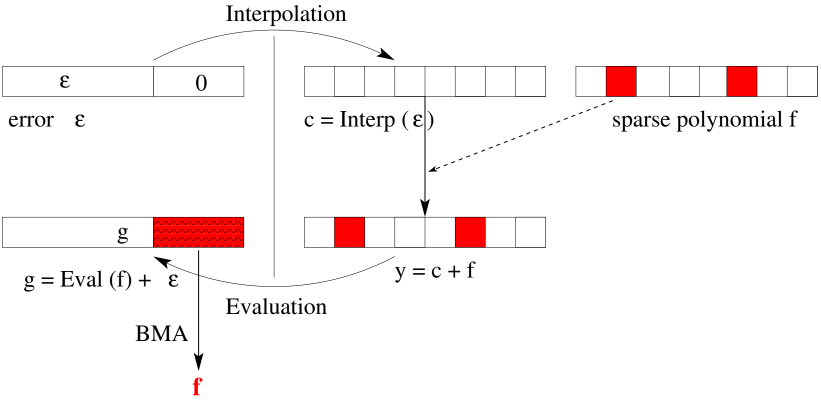
Sparse interpolation with errors

Find f from $(f(w^1), \ldots, f(w^n)) + \varepsilon$



Sparse interpolation with errors

Find f from $(f(w^1), \ldots, f(w^n)) + \varepsilon$



Outline

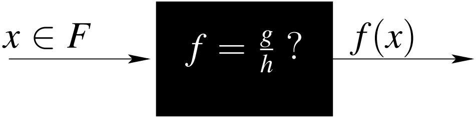
Dense polynomial interpolation with errors

Sparse polynomial interpolation with errors

- de Proni/Ben-Or/Tiwari interpolation
- Fault tolerant Berlekamp/Massey algorithm
- Relations to Reed-Solomon decoding

Dense rational function interpolation with errors

Dense rational function interpolation

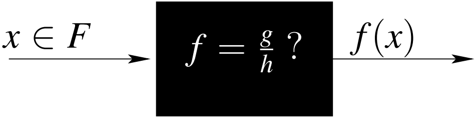


$$g = \sum_{i=0}^{d_G} g_i x^i, \quad h = \sum_{i=0}^{d_H} h_i x^i$$

Problem

Recover $g, h \in K[X]$,*with* $\deg g \leq d_G, \deg h \leq d_H$. *given evaluations of* $f = \frac{g}{h}$.

Dense rational function interpolation



$$g = \sum_{i=0}^{d_G} g_i x^i, \quad h = \sum_{i=0}^{d_H} h_i x^i$$

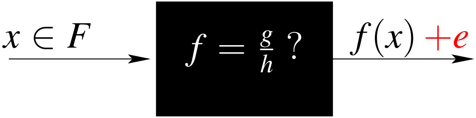
Problem

Recover $g, h \in K[X]$,*with* $\deg g \leq d_G, \deg h \leq d_H$. *given evaluations of* $f = \frac{g}{h}$.

Cauchy interpolation

$\Rightarrow$only $d_F + d_G + 1$ entries needed

Dense rational function interpolation with errors



$$g = \sum_{i=0}^{d_G} g_i x^i, \quad h = \sum_{i=0}^{d_H} h_i x^i$$

Problem

Recover $g, h \in K[X]$,*with* $\deg g \leq d_G, \deg h \leq d_H$. *given evaluations of* $f = \frac{g}{h}$.

Cauchy interpolation

$\Rightarrow$only $d_F + d_G + 1$ entries needed

[Kaltofen and Pernet, 2013]

$\Rightarrow$only $d_F + d_G + 2E + 1$ evaluations needed with E errors.

$\Rightarrow$smoothly supports evaluations at poles and erroneous poles

Rational function reconstruction

Problem (RFR: Rational Function Reconstruction)

Given $A, B \in K[X]$ *with* $\deg B < \deg A = n$, *recover* $g, h \in K[X]$, *with* $\deg g \leq d_G, \deg h \leq n - d_G - 1$ *and*

$$g = hB \mod A.$$

Theorem

Let $(f_0 = A, f_1 = B, \ldots, f_\ell)$ *be the sequence of remainders in the Ext. Euclidean alg. applied on* (A, B) *and* u_i, v_i *the multipliers s.t.* $f_i = u_i f_0 + v_i f_1$. *Then at iteration* j *s.t.* $\deg f_j \leq d_G < \deg f_{j-1}$,

1. (f_j, v_j) *is a solution to the RFR problem.*
2. *it is minimal: any other solution* (g, h) *is of the form* $g = q f_j$, $h = q v_j$.

Instantiations

Dense polynomial interpolation with errors

- Erroneous interpolant: $P = \text{Interp}((y_i, x_i))$
- Error locator polynomial:$\Lambda = \prod_{i | y_i \text{is erroneous}} (X - x_i)$

Find f with $\deg f \leq d_F$ s.t. f and H agree on at least $n - t$ evaluations x_i.

$$\underbrace{\Lambda f}_{f_j} = \underbrace{\Lambda}_{g_j} P \mod \prod_{i=1}^{n} (X - x_i)$$

and $(\Lambda f, \Lambda)$ is minimal.

$\Rightarrow$computed by Ext. Euclidean Algorithm

$$f = f_j / g_j.$$

Instantiations

Cauchy interpolation

- Polynomial interpolant: $P = \text{Interp}((y_i, x_i))$

Find g, h with $\deg g \leq d_G$ $\deg h \leq n - d_G - 1$ s.t. $\frac{g}{h} = P$ $\bmod \prod_{i=1}^{n}(X - x_i)$.

$$\underbrace{g}_{f_j} = \underbrace{h}_{g_j} P \mod \prod_{i=1}^{n}(X - x_i)$$

and (g, h) is minimal.
$\Rightarrow$computed by Ext. Euclidean Algorithm

$$\frac{g}{h} = \frac{f_j}{g_j}.$$

Instantiations

Cauchy interpolation at poles (with multiplicity 1)

- value at a pole $\rightarrow \infty$.
- Pole locator: $P_\infty = \prod_{i|y_i=\infty}(X - x_i)$
- $h = \overline{h}P_\infty$
- Polynomial interpolant: $P = \text{Interp}((y_i, x_i)\text{for} y_i \neq \infty)$

$$\underbrace{g}_{f_j} = \underbrace{\overline{h}}_{g_j} P \mod \prod_{i=1}^{n}(X - x_i)/P_\infty$$

and $(g, \overline{h})$ is minimal.
$\Rightarrow$computed by Ext. Euclidean Algorithm

$$\frac{g}{h} = \frac{f_j}{g_j P_\infty}.$$

Instantiations

Cauchy interpolation at poles with errors

- value at a pole $\to \infty$.
- Pole locator: $P_\infty = \prod_{i|y_i=\infty}(X - x_i) = \underbrace{G_\infty}_{\text{true poles}} \underbrace{\Lambda_\infty}_{\text{erroneous poles}}$
- $h = \overline{h}P_\infty$
- Polynomial interpolant: $P = \text{Interp}((y_iP_\infty(x_i), x_i) \text{for} y_i \neq \infty)$
- Error locator polynomial: $\Lambda = \prod_{i|y_i \text{is erroneous}}(X - x_i) = \overline{\Lambda}\Lambda_\infty$

$$\underbrace{g\Lambda P_\infty}_{f_j} = \underbrace{\overline{h\Lambda}}_{g_j} PP_\infty \mod \prod_{i=1}^{n}(X - x_i)$$

and $(g\Lambda P_\infty, \overline{h\Lambda})$ is minimal.
$\Rightarrow$computed by Ext. Euclidean Algorithm

$$\frac{g}{h} = \frac{f_j}{g_jP_\infty^2}.$$

Thank you

References

Ben-Or, M. and Tiwari, P. (1988).
A deterministic algorithm for sparse multivariate polynomial interpolation.
In *STOC'88*, pages 301–309.

Blahut, R. E. (1984).
A universal Reed-Solomon decoder.
IBM J. Res. Develop., 18(2):943–959.

Comer, M. T., Kaltofen, E. L., and Pernet, C. (2012).
Sparse polynomial interpolation and Berlekamp/Massey algorithms that correct outlier errors in input values.
In *ISSAC '12*, pages 138–145.

Kaltofen, E. and Pernet, C. (2013).
Cauchy interpolation with errors in the values.
Manuscript in preparation.

Khonji, M., Pernet, C., Roch, J.-L., Roche, T., and Stalinsky, T. (2010).
Output-sensitive decoding for redundant residue systems.
In *ISSAC'10*, pages 265–272.