Fast algorithms for multivariate interpolation problems

Vincent Neiger[§,†,‡]

Claude-Pierre Jeannerod[§] Éric Schost[†] Gilles Villard[§]

[§]AriC, LIP, École Normale Supérieure de Lyon, France

[†]ORCCA, Computer Science Department, Western University, London, ON, Canada

[‡]Supported by the international mobility grant *Explo'ra doc* from *Région Rhône-Alpes*

PolSys Seminar
LIP6, Paris, France, July 2, 2015



Outline

1. List-decoding and interpolation problem
2. Fast algorithms using structured linear algebra
3. Fast algorithms using linear algebra over $\mathbb{K}[X]$

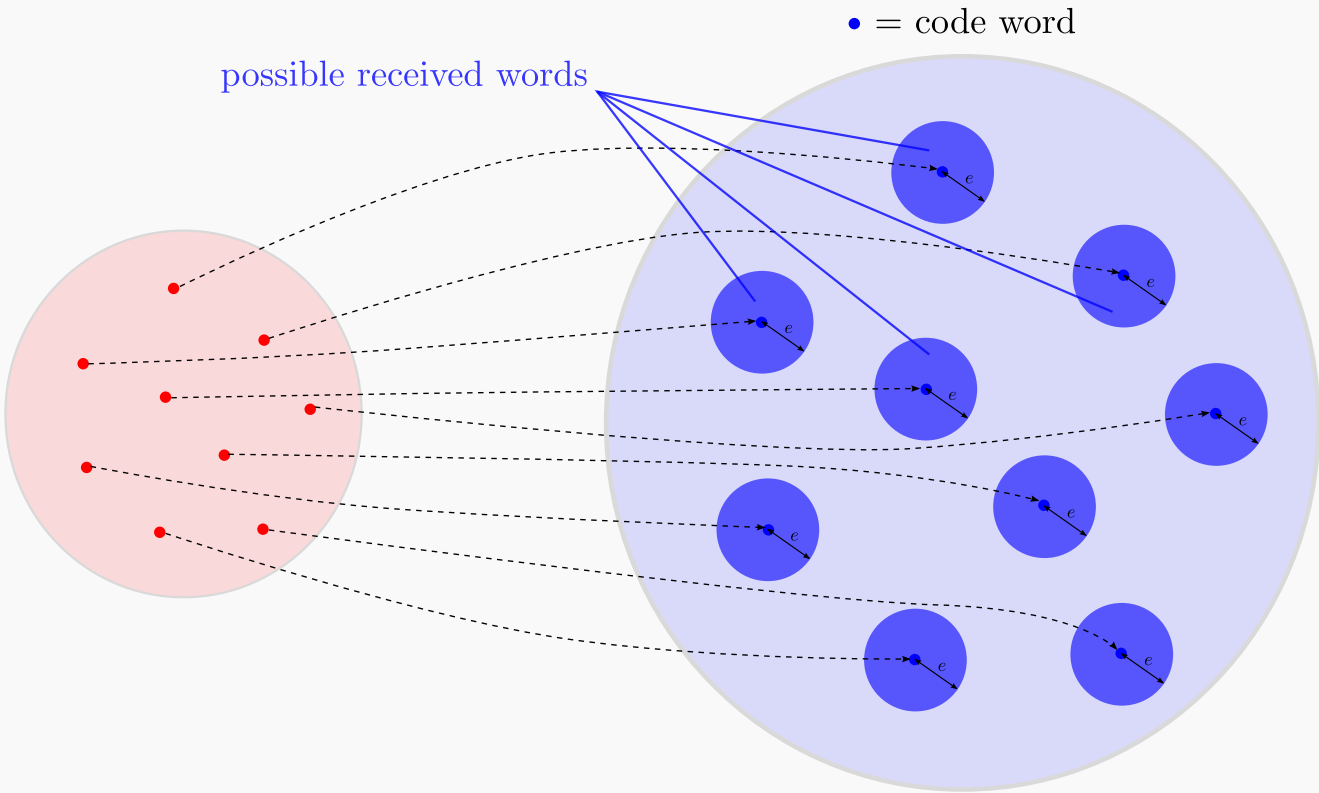
Outline

Reed-Solomon codes

Reliable delivery of data over an unreliable communication channel

$$w = w_0 + \cdots + w_k X^k \xrightarrow{\text{encoding}} (w(x_1), \ldots, w(x_n)) \xrightarrow{\text{noise}} y = (y_1, \ldots, y_n)$$

At most e errors during transmission: $\#\{i \mid w(x_i) \neq y_i\} \leqslant e$



Unique decoding problem

$$w = w_0 + \cdots + w_k X^k \xrightarrow{\text{encoding}} (w(x_1), \ldots, w(x_n)) \xrightarrow{\text{noise}} y = (y_1, \ldots, y_n)$$

where $\#\{i \mid w(x_i) \neq y_i\} \leqslant e$

$\rightsquigarrow$ the receiver gets y and looks for w

Unique decoding of Reed-Solomon codes

Input:

$x_1, \ldots, x_n$ the n distinct evaluation points in $\mathbb{K}$,
k the degree bound, e the error-correction radius,
$(y_1, \ldots, y_n)$ the received word in $\mathbb{K}^n$

Unique decoding assumption: $e < \frac{n-k}{2}$

Output:

The polynomial w in $\mathbb{K}[X]$ such that

$$\deg w \leqslant k \qquad \text{and} \qquad \#\{i \mid w(x_i) \neq y_i\} \leqslant e.$$

Key equations (unique decoding)

Define the interpolation polynomial

$$R(X) \text{ such that } R(x_i) = y_i,$$

and the error-locator polynomial

$$\Lambda(X) = \prod_{i \mid \text{error}} (X - x_i).$$

$\Lambda(X)$ is an unknown polynomial with $\deg \Lambda \leqslant e$

Key equations

$$\text{for every } i, \quad \Lambda(x_i) R(x_i) = \Lambda(x_i) w(x_i)$$

Quadratic equations in the unknown coefficients of w and Λ...

Modular key equation (unique decoding)

Recall the interpolation and error-locator polynomials

$$R(x_i) = y_i, \qquad \Lambda(X) = \prod_{i \mid \text{error}} (X - x_i)$$

Key equations

$$\text{for every } i, \quad \Lambda(x_i)R(x_i) = \Lambda(x_i)w(x_i)$$

i.e. for every i, $\quad \Lambda(X)R(X) = \Lambda(X)w(X) \mod (X - x_i)$

Modular key equation (unique decoding)

Recall the interpolation and error-locator polynomials

$$R(x_i) = y_i, \qquad \Lambda(X) = \prod_{i \mid \text{error}} (X - x_i)$$

Key equations

$$\text{for every } i, \quad \Lambda(x_i)R(x_i) = \Lambda(x_i)w(x_i)$$

i.e. for every i, $\Lambda(X)R(X) = \Lambda(X)w(X) \mod (X - x_i)$

Define the master polynomial

$$G(X) = \prod_{1 \leqslant i \leqslant n} (X - x_i)$$

Modular key equation

$$\Lambda(X)R(X) = \Lambda(X)w(X) \mod G(X)$$

Fast algorithm via rational function reconstruction

Modular key equation:

$$\Lambda R \;=\; \Lambda w \mod G$$

$\Longrightarrow \lambda = \Lambda, \omega = \Lambda w$ solution of the rational reconstruction problem

$$\begin{cases} \lambda R = \omega \mod G, \\ \deg(\lambda) \leqslant e, \quad \deg(\omega) \leqslant e + k, \quad \lambda \text{ monic.} \end{cases}$$

Fast algorithm via rational function reconstruction

Modular key equation:

$$\Lambda R = \Lambda w \mod G$$

$\Longrightarrow \lambda = \Lambda, \omega = \Lambda w$ solution of the rational reconstruction problem

$$\begin{cases} \lambda R = \omega \mod G, \\ \deg(\lambda) \leqslant e, \quad \deg(\omega) \leqslant e + k, \quad \lambda \text{ monic.} \end{cases}$$

Unique decoding assumption: $e + k < n - e$
$\Longrightarrow$ unique rational solution $\frac{\omega}{\lambda} = \frac{\Lambda w}{\Lambda} = w$

cost bound $\tilde{\mathcal{O}}(n)$ using extended Euclidean algorithm

[Modern Computer Algebra, von zur Gathen - Gerhard, 2013]

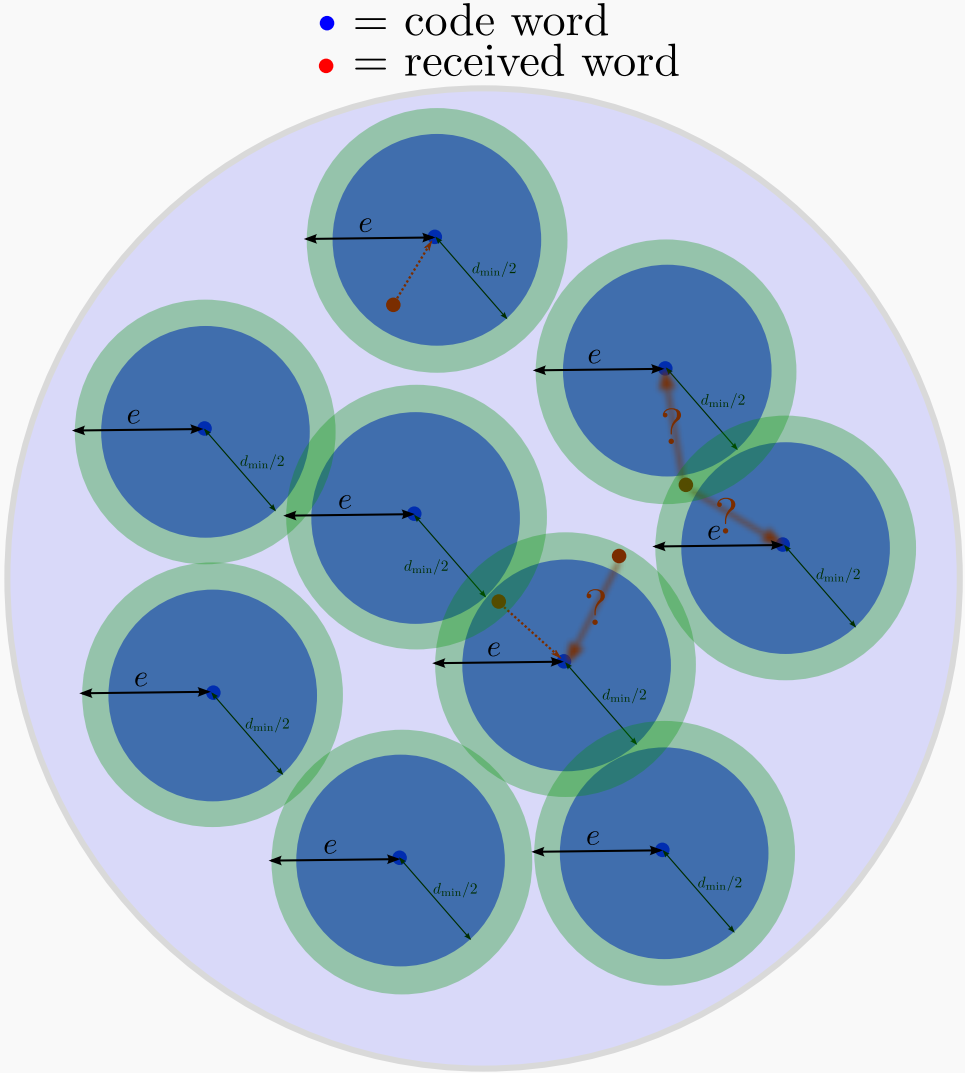
Non-unique decoding

How to "decode" when more errors?

transmission with $\leqslant e$ errors
where $e \geqslant d_{\min}/2$

possibly two (or more) code words
at the same distance...

the closest code word is not necessarily the one which was sent...



Non-unique decoding

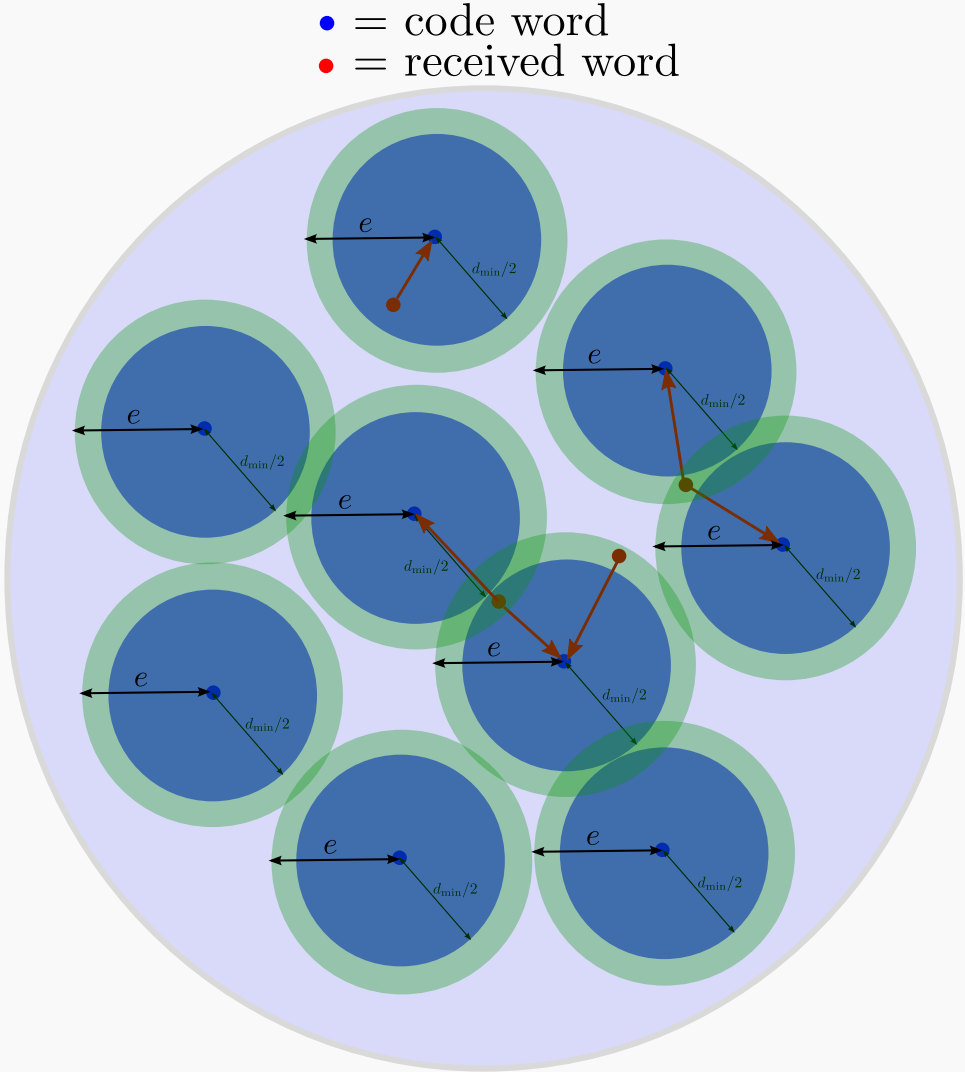
How to "decode" when more errors?

transmission with $\leqslant e$ errors
where $e \geqslant d_{\min}/2$

possibly two (or more) code words
at the same distance...

the closest code word is not necessarily the one which was sent...

$\Rightarrow$ Return a list of all code words
at distance $\leqslant e$
(called list-decoding)



List-decoding problem

List-decoding Reed-Solomon codes

Input:

$x_1, \ldots, x_n$ the n distinct evaluation points in $\mathbb{K}$,
k the degree bound, e the error-correction radius,
$(y_1, \ldots, y_n)$ the received word in $\mathbb{K}^n$

List-decoding assumption: $e < n - \sqrt{kn}$ [Guruswami - Sudan 1999]

Output:

list of all polynomials w in $\mathbb{K}[X]$ such that

$$\deg w \leqslant k \qquad \text{and} \qquad \#\{i \mid w(x_i) \neq y_i\} \leqslant e.$$

Towards the interpolation problem (1/3)

For one solution w_1, modular key equation

$$\Lambda_1 R \;=\; \Lambda_1 w_1 \mod G$$

where
$R(x_i) = y_i$, $G(X) = \prod_{1 \leqslant i \leqslant n}(X - x_i)$, $\Lambda_1(X) = \prod_{i \mid \text{error}_1}(X - x_i)$.

Possibly $\deg(\Lambda_1) + \deg(\Lambda_1 w_1) \geqslant n = \deg G$
$\Longrightarrow$ no uniqueness of a rational solution to $\lambda_1 R \;=\; \omega_1 \mod G$

Towards the interpolation problem (2/3)

Key equation

$$\Lambda_1(R - w_1) \;=\; 0 \mod G$$

For two solutions w_1 and w_2, key equation

$$\Lambda(R - w_1)(R - w_2) \;=\; 0 \mod G$$

where $\Lambda = \prod_{i \mid \text{error}_{1\wedge 2}}(X - x_i) = \gcd(\Lambda_1, \Lambda_2)$.

$\Longrightarrow$ w_1, w_2 are Y-roots of the bivariate polynomial

$$Q(X, Y) \;=\; \Lambda(Y - w_1)(Y - w_2)$$

Towards the interpolation problem (3/3)

$\Lambda(R - w_1)(R - w_2) = 0 \mod G$
w_1, w_2 are Y-roots of

$$\begin{aligned} Q(X, Y) &= \Lambda(Y - w_1)(Y - w_2) \\ &= \Lambda w_1 w_2 - \Lambda(w_1 + w_2)Y + \Lambda Y^2 \end{aligned}$$

Similar properties for all solutions $w_1, \ldots, w_\ell$

Properties of $Q(X, Y)$:

- $\deg_Y Q$ is the number of solutions ℓ
- coefficients in X of Q have small degree
- key equation $Q(X, R) = 0 \mod G$
  that is, $Q(x_i, y_i) = 0$ for every i

List-decoding: Guruswami-Sudan algorithm

w solution: $\deg w \leqslant k$ and $\#\{i \mid w(x_i) \neq y_i\} \leqslant e$

[Guruswami - Sudan, 1999]

- Interpolation step
  compute a polynomial $Q(X, Y)$ such that:
  - $Q(X, w)$ has small degree
  - $Q(X, w)$ has many roots

  $\longrightarrow$ w solution $\Rightarrow Q(X, w) = 0$
- Root-finding step
  find all Y-roots of $Q(X, Y)$, keep those that are solutions

Here we focus on the Interpolation step.

The interpolation problem

Bivariate interpolation

Input: points $\{(x_i, y_i)\}_{1 \leqslant i \leqslant n}$, number of points n, degree weight k, weighted-degree bound b, list size ℓ

Output: a nonzero polynomial Q in $\mathbb{K}[X, Y]$ such that

(i)	$\deg_Y Q \leqslant \ell$,	(list-size condition)
(ii)	$\deg_X Q(X, X^k Y) < b$,	(weighted-degree condition)
(iii)	for $1 \leqslant i \leqslant n$, $Q(x_i, y_i) = 0$	(vanishing condition)

[Guruswami - Sudan, 1999]:
$e < n - \sqrt{kn} \Rightarrow$ solution exists for some well-chosen ℓ
Using Gaussian elimination, cost $\mathcal{O}(n^\omega)$ ($\omega =$ exponent of mat. mult.)

Note: in general $e < n - \sqrt{kn}$ requires vanishing with some multiplicity μ.
Here, for simplicity, $\mu = 1$.

Outline

Overview

[Olshevsky - Shokrollahi, 1999]
linearize the vanishing condition on each point
Vandermonde-like system, cost $\mathcal{O}(\ell n^2)$

[Roth - Ruckenstein, 2000] [Zeh - Gentner - Augot, 2011]
linearize the reversed extended key equation
Mosaic-Hankel system, cost $\mathcal{O}(\ell n^2)$
using an adapted version of [Feng - Tzeng, 1991]

[Chowdhury - Jeannerod - Neiger - Schost - Villard, 2015]
linearize the extended key equation
Toeplitz-like system, cost $\tilde{\mathcal{O}}(\ell^{\omega-1} n)$
using [Bostan - Jeannerod - Schost, 2007]

Extended Key Equation

[Roth - Ruckenstein, 2000] [Zeh - Gentner - Augot, 2011]

Assuming x_i pairwise distinct,

vanishing condition: $Q(x_i, y_i) = 0$ for $i \in \{1, \ldots, n\}$

$\iff$ *extended* key equation: $Q(X, R) = 0 \mod G$

where $G = \prod_{1 \leqslant i \leqslant n}(X - x_i)$ and $\forall i, R(x_i) = y_i$.

List-size condition: linearize over $\mathbb{K}[X]$,

$$Q(X, Y) = \sum_{0 \leqslant j \leqslant \ell} Q_j(X) Y^j$$

Weighted-degree condition: $\deg Q_j(X) < b - jk$.

Polynomial approximation problem

Vanishing condition + list-size condition + weighted-degree condition

$$\Longleftrightarrow \begin{cases} \displaystyle\sum_{0 \leqslant j \leqslant \ell} Q_j(X) R(X)^j \;=\; 0 \mod G(X) \\ \deg Q_j(X) < b - jk \quad \text{for } j \leqslant \ell \end{cases}$$

In [Roth - Ruckenstein, 2000] and [Zeh - Gentner - Augot, 2011]

- reverse all polynomials

$$\Longleftrightarrow \begin{cases} \displaystyle\sum_{0 \leqslant j \leqslant \ell} \overline{Q_j(X)} S_j(X) \;=\; \overline{B(X)} \mod X^{n+b-1} \\ \deg \overline{Q_j(X)} < b - jk \quad \text{for } j \leqslant \ell \end{cases}$$

- linearize as a mosaic-Hankel system over $\mathbb{K}$,
- find a solution in $\mathcal{O}(\ell n^2)$ using an adapted [Feng - Tzeng, 1991]

Direct linearization with a companion matrix (1/2)

[Chowdhury - Jeannerod - Neiger - Schost - Villard, 2015]

Write $Q_j(X) = \sum_{r<b-jk} Q_j^{(r)} X^r$, then the equation becomes

$$\sum_{0 \leqslant j \leqslant \ell} \sum_{r<b-jk} Q_j^{(r)} X^r \underbrace{R(X)^j}_{F_j(X)} = 0 \mod G(X)$$

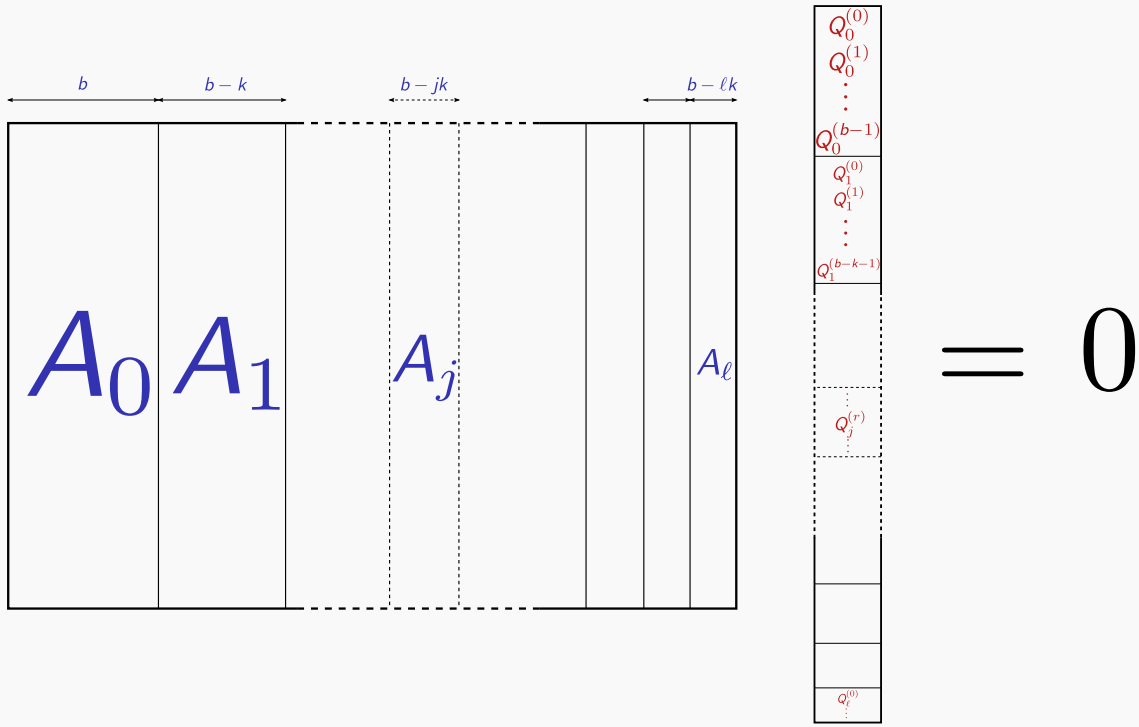
Define the companion matrix

$$\mathcal{C}(G) = \begin{bmatrix} 0 & 0 & \cdots & 0 & -G_0 \\ 1 & 0 & \cdots & 0 & -G_1 \\ 0 & 1 & \cdots & 0 & -G_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -G_{n-1} \end{bmatrix} \in \mathbb{K}^{n \times n}$$

Key property:
multiplication by $\mathcal{C}(G)$ on the left is multiplication by X modulo $G(X)$

Direct linearization (2/2)

Solution $\Longleftrightarrow$ nonzero vector in the nullspace of $A = [A_0 | \cdots | A_\ell]$



where the block $A_j \in \mathbb{K}^{n \times b-jk}$ is defined by its first column

$$c^{(0)} = \begin{bmatrix} F_j^{(0)} \\ \vdots \\ F_j^{(n-1)} \end{bmatrix}$$

and the subsequent columns $c^{(r+1)} = \mathcal{C}(G) \cdot c^{(r)}$

Displacement rank

A is $n \times u$ where $u = b + (b-k) + (b-2k) + \cdots + (b-\ell k)$
$n =$ number of linear equations $Q(x_i, y_i) = 0$
$u =$ number of unknown coefficients of $Q(X, Y)$

$$\mathcal{Z}_n = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix} \in \mathbb{K}^{n \times n}$$

displacement operator
$A \mapsto A - \mathcal{Z}_n A \mathcal{Z}_u^T$
$\rightsquigarrow$ Toeplitz structure

Fact: $A - \mathcal{Z}_n A \mathcal{Z}_u^T$ has rank $\leqslant \ell + 2$

Conclusion: Toeplitz-like system with displacement rank $\leqslant \ell + 2$

Complexity bound for this approach

Solving the structured linear system [Bitmead - Anderson, 1980] [Morf, 1980] [Kaltofen, 1994] [Pan, 2001] [Bostan - Jeannerod - Schost, 2007]

Two main operations:

- computing generators (compact representation)
  $\approx$ computing the first column, last column, first row of each block
  $\rightsquigarrow$ cost $\tilde{\mathcal{O}}(\ell n)$
- solving the system
  number of equations n, displacement rank $\leqslant \ell + 2$
  $\rightsquigarrow$ cost $\tilde{\mathcal{O}}(\ell^{\omega-1} n)$, probabilistic algorithm

Structured system approach:

$$\tilde{\mathcal{O}}(\ell^{\omega-1} n), \quad \text{probabilistic algorithm}$$

Goal: with same cost bound,

- deterministic algorithm?
- accepting repeated x_i?

Outline

Bases of interpolants

Recall we have points $\{(x_i, y_i)\}_{1 \leqslant i \leqslant n}$, weight k, degree bounds ℓ and b

- the set of interpolants

$$\{(Q_0, \ldots, Q_\ell) \in \mathbb{K}[X]^{\ell+1} / Q(x_i, y_i) = 0 \text{ for } 1 \leqslant i \leqslant n\}$$

  is a free $\mathbb{K}[X]$-module of rank $\ell + 1$
- *interpolation basis* = basis of this module
  $\rightsquigarrow$ matrix P in $\mathbb{K}[X]^{\ell+1 \times \ell+1}$ with rows $P_0, \ldots, P_\ell$ interpolants
- P is a *k-minimal* interpolation basis if
  $(\deg_X P_0(X, X^k Y), \ldots, \deg_X P_\ell(X, X^k Y))$
  is lexicographically minimal

$\Longrightarrow$ an interpolant Q such that $\deg_X Q(X, X^k Y) < b$ can be found in a k-minimal interpolation basis (unless no such Q exists)

Using polynomial lattice reduction

[Alekhnovich, 2002] [Reinhard, 2003] [Beelen - Brander, 2010]
[Bernstein, 2011] [Cohn - Heninger, 2011] [Cohn - Heninger, 2012]

- Compute a known basis of interpolants

$$\begin{array}{ccc} G & \longrightarrow & \\ Y-R & \longrightarrow & \\ Y(Y-R) & \longrightarrow & \\ \vdots & \vdots & \\ Y^{\ell}(Y-R) & \longrightarrow & \end{array} \begin{bmatrix} G & 0 & \cdots & 0 & 0 \\ -R & 1 & \cdots & 0 & 0 \\ 0 & -R & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -R & 1 \end{bmatrix} \in \mathbb{K}[X]^{\ell+1 \times \ell+1}$$

- Lattice basis reduction $\rightsquigarrow$ k-minimal interpolation basis
- Return row with smallest weighted-degree

Fastest known cost: $\tilde{\mathcal{O}}(\ell^{\omega} n)$ [Bernstein, 2011] [Cohn - Heninger, 2011]
using deterministic reduction [Gupta - Sarkar - Storjohann - Valeriote, 2012]

Using order basis computation

- Derive extended key equation as in the structured approach
  Vanishing condition + list-size condition + weighted-degree condition

$$\iff \begin{cases} \displaystyle\sum_{0 \leqslant j \leqslant \ell} Q_j(X) F_j(X) \;=\; 0 \mod G(X) \\ \deg(Q_0, X^k Q_1, \ldots, X^{\ell k} Q_\ell) < b \end{cases}$$

- Reverse it $\rightsquigarrow$ Hermite-Padé approximation

$$\iff \begin{cases} \displaystyle\sum_{0 \leqslant j \leqslant \ell} \overline{Q_j(X)} S_j(X) \;=\; \overline{B(X)} \mod X^{n+b-1} \\ \deg(\overline{Q_0}, X^k \overline{Q_1}, \ldots, X^{\ell k} \overline{Q_\ell}, -X\overline{B}) < b \end{cases}$$

Cost $\tilde{\mathcal{O}}(\ell^{\omega-1} n)$ using [Zhou - Labahn, 2012], deterministic

Structured system approach:

$$\tilde{\mathcal{O}}(\ell^{\omega-1} n), \quad \text{probabilistic algorithm}$$

Order basis approach:

$$\tilde{\mathcal{O}}(\ell^{\omega-1} n), \quad \text{deterministic algorithm}$$

Goal: with same cost bound,
deterministic algorithm accepting repeated x_i?

Iterative algorithm

Sometimes called Kötter algorithm in coding theory.
Related literature: [Feng - Tzeng, 1991] [Beckermann - Labahn, 1994 & 2000] [Kötter, 1996] [Nielsen - Høholdt, 1998]

Algorithm (`Interpolation`)

1. $P = \text{identity}$
2. weights $\vec{w} = (0, k, 2k, \ldots, \ell k)$
3. *For* i *from* 1 *to* n *do*
   - **a.** Evaluate: compute $P_0(x_i, y_i), \ldots, P_\ell(x_i, y_i)$
   - **b.** Choose pivot: π with smallest w_π such that $P_\pi(x_i, y_i) \neq 0$; $w_\pi = w_\pi + 1$
   - **c.** Eliminate:
     For $j \neq \pi$ *do* $P_j = P_j - \frac{P_j(x_i,y_i)}{P_\pi(x_i,y_i)} P_\pi$ /* $\forall j \neq \pi,\ P_j(x_i, y_i) = 0$ */
     $P_\pi = (X - x_i) P_\pi$ /* $P_\pi(x_i, y_i) = 0$ */

After i iterations P is a $\vec{w}$-minimal interpolation basis for points $1, \ldots, i$

Weights and size of the basis

Iterative algorithm:

- Cost bound: $\mathcal{O}(\ell n^2 + \ell^2 n) \subseteq \mathcal{O}(\ell n^2)$ when $\ell \leqslant n$
- accepts repeated x_i

Goal: divide-and-conquer version in $\mathcal{O}^{\sim}(\ell^{\omega-1} n)$

In general, the size of a $\vec{w}$-minimal basis is $\mathcal{O}(\ell^2 n)$
$\rightsquigarrow$ leaves no hope for a $\mathcal{O}^{\sim}(\ell^{\omega-1} n)$ algorithm...

Weights and size of the basis

Iterative algorithm:

- Cost bound: $\mathcal{O}(\ell n^2 + \ell^2 n) \subseteq \mathcal{O}(\ell n^2)$ when $\ell \leqslant n$
- accepts repeated x_i

Goal: divide-and-conquer version in $\mathcal{O}^{\sim}(\ell^{\omega-1} n)$

In general, the size of a $\vec{w}$-minimal basis is $\mathcal{O}(\ell^2 n)$
$\rightsquigarrow$ leaves no hope for a $\mathcal{O}^{\sim}(\ell^{\omega-1} n)$ algorithm...

Assumption: weight $\vec{w}$ satisfies $w_0 + \cdots + w_\ell \in \mathcal{O}(n)$
(satisfied in coding theory applications)

Consequence: a $\vec{w}$-minimal basis has sum of row degrees $\mathcal{O}(n)$
$\rightsquigarrow$ in particular, size $\mathcal{O}(\ell n)$

Ingredient 1: recursion base case

Base case: $\ell = n$, goal: $\tilde{\mathcal{O}}(\ell^\omega)$

Complete linearization over $\mathbb{K}$
(similar to linearization in [Beckermann - Labahn, 2000])

minimal interpolation basis $\longleftrightarrow$ minimal linear relations between rows

fast computation of those relations, in $\mathcal{O}(\ell^\omega \log \ell)$

Ingredient 2: evaluation matrix

Divide-and-conquer:

- $P^{(1)}$ for first $n/2$ points
- $P^{(2)}$ for last $n/2$ points with $P^{(1)}$-dependent evaluation
- return $P^{(2)}P^{(1)}$

Store an evaluation matrix E
= evaluations of the global basis at points currently processed

Ingredient 2: evaluation matrix

Divide-and-conquer:

- $P^{(1)}$ for first $n/2$ points
- $P^{(2)}$ for last $n/2$ points with $P^{(1)}$-dependent evaluation
- return $P^{(2)}P^{(1)}$

Store an evaluation matrix E
= evaluations of the global basis at points currently processed

Example: after first recursive call,

$$E = \begin{bmatrix} 0 & \cdots & 0 & P_0^{(1)}(x_{n/2+1}, y_{n/2+1}) & \cdots & P_0^{(1)}(x_n, y_n) \\ 0 & \cdots & 0 & P_1^{(1)}(x_{n/2+1}, y_{n/2+1}) & \cdots & P_1^{(1)}(x_n, y_n) \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ 0 & \cdots & 0 & P_\ell^{(1)}(x_{n/2+1}, y_{n/2+1}) & \cdots & P_\ell^{(1)}(x_n, y_n) \end{bmatrix} \in \mathbb{K}^{\ell+1 \times n}$$

Using assumption on the weight,

update of the evaluation matrix is computed in $\tilde{\mathcal{O}}(\ell^{\omega-1} n)$

Divide-and-conquer algorithm

Central result: [Beckermann - Labahn, 1994 & 2000]
transitivity of interpolation bases

Algorithm (Fast bivariate interpolation)

- $P^{(1)}$ for first $n/2$ points, evaluations E, and weights $\vec{w}$
- E' updated evaluations, $\vec{w}'$ updated weight
- $P^{(2)}$ for last $n/2$ points, evaluations E', weights $\vec{w}'$
- return $P = P^{(2)} P^{(1)}$

P is a $\vec{w}$-minimal interpolation basis

Divide-and-conquer algorithm

Central result: [Beckermann - Labahn, 1994 & 2000]
transitivity of interpolation bases

Algorithm (Fast bivariate interpolation)

- $P^{(1)}$ for first $n/2$ points, evaluations E, and weights $\vec{w}$
- E' updated evaluations, $\vec{w}'$ updated weight
- $P^{(2)}$ for last $n/2$ points, evaluations E', weights $\vec{w}'$
- return $P = P^{(2)}P^{(1)}$

P is a $\vec{w}$-minimal interpolation basis

Using assumption on the weight,

$$P = P^{(2)}P^{(1)} \text{ is computed in } \tilde{\mathcal{O}}(\ell^{\omega-1}n)$$

Ingredient 3: assumption on the weights

When $w_0 + \cdots + w_\ell \in \mathcal{O}(n)$,
complexity bound equation: $\mathcal{C}(\ell, n) = 2\mathcal{C}(\ell, n/2) + \mathcal{O}^{\sim}(\ell^{\omega-1} n)$?

Ingredient 3: assumption on the weights

When $w_0 + \cdots + w_\ell \in \mathcal{O}(n)$,
complexity bound equation: $\mathcal{C}(\ell, n) = 2\mathcal{C}(\ell, n/2) + \mathcal{O}^{\sim}(\ell^{\omega-1} n)$?

Problem:
In the recursive call, n becomes $n/2$ but $\vec{w}$ is unchanged

Preserve the assumption on $\vec{w}$ throughout recursive calls

- Compute a minimal interpolation basis for $\vec{w} = \vec{0}$
- Recover a $\vec{w}$-minimal basis by change of weights
- Done in $\mathcal{O}^{\sim}(\ell^{\omega-1} n)$ by computing a minimal nullspace basis [Zhou Labahn Storjohann, 2012]

Conclusion

Bivariate interpolation algorithm:

- Cost bound $\tilde{\mathcal{O}}(\ell^{\omega-1} n)$
- Deterministic
- No assumption on x_i
- Returns a minimal interpolation basis

With the same algorithm:

- Multiplicity μ_i on each point (x_i, y_i)
  (e.g. for soft-decoding of Reed-Solomon codes)
- Several variables $Q(X, Y_1, \ldots, Y_s)$
  (e.g. for folded Reed-Solomon codes,
  or Private Information Retrieval)

Remark: Does fast order basis as a special case