Sparse Polynomial Interpolation and Berlekamp/Massey algorithms that correct Outlier Errors in Input Values

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

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

ISSAC'12, Grenoble, France,
July 23rd, 2012

Outline

- Berlekamp/Massey algorithm with errors
  - Bounds on the decoding capacity
  - Decoding algorithms
- Sparse Polynomial Interpolation with errors
- Relations to Reed-Solomon decoding

Introduction

Polynomial interpolation with errors

Dense case: Reed-Solomon codes / CRT codes
$\Rightarrow$number of evaluation points made adaptive on error impact and degree [Khonji & Al.'10]

Introduction

Polynomial interpolation with errors

Dense case: Reed-Solomon codes / CRT codes
⇒number of evaluation points made adaptive on error impact and degree [Khonji & Al.'10]

Sparse case: Present work

- based on Ben-Or & Tiwari's interpolation algorithm
- itself based on Berlekamp/Massey algorithm
  ⇒develop Berlekamp/Massey Algorithm with errors

Preliminaries

Linear recurring sequences

Sequence $(a_0, a_1, \ldots, a_n, \ldots)$ such that

$$\forall j \geq 0 \; a_{j+t} = \sum_{i=0}^{t-1} \lambda_i a_{i+j}$$

generating polynomial: $\Lambda(z) = z^t - \sum_{i=0}^{t-1} \lambda_i z^i$

minimal generating polynomial: $\Lambda(z)$ of minimal degree

linear complexity of $(a_i)_i$: the minimal degree of Λ

Hamming weight: $\text{weight}(x) = \#\{i | x_i \neq 0\}$

Hamming distance: $d_H(x, y) = \text{weight}(x - y)$

Berlekamp/Massey algorithm

Input: $(a_0, \ldots, a_{n-1})$ a sequence of field elements.
Result: $\Lambda(z) = \sum_{i=0}^{L_n} \lambda_i z^i$ a monic polynomial of minimal degree $L_n \leq n$ such that $\sum_{i=0}^{L_n} \lambda_i a_{i+j} = 0$ for $j = 0, \ldots, n - L_n - 1$.

- Guarantee : BMA finds Λ of degree t from $\leq 2t$ entries.

Outline

Berlekamp/Massey algorithm with errors

- Bounds on the decoding capacity
- Decoding algorithms

Sparse Polynomial Interpolation with errors

Relations to Reed-Solomon decoding

Problem Statement

Berlkamp/Massey with errors

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 including $(a_0, a_1, \dots)$

Problem Statement

Berlkamp/Massey with errors

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 including $(a_0, a_1, \dots)$

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 $E = 1, t = 2$

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

Where is the error?

How many entries to guarantee uniqueness?

Case $E = 1, t = 2$

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

Where is the error?

How many entries to guarantee uniqueness?

Case $E = 1, t = 2$

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

Where is the error?

How many entries to guarantee uniqueness?

Case $E = 1, t = 2$

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

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

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

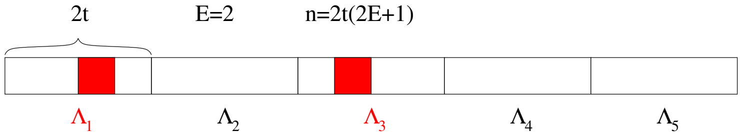
is generated by $z^t - 1$ or $z^t + 1$ up to E errors.
$\Rightarrow$ambiguity with $n = 2t(2E + 1) - 1$ values.

Theorem

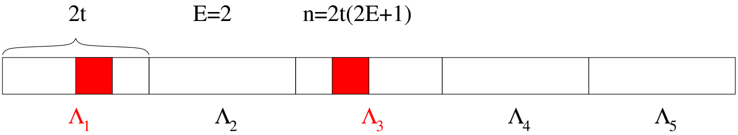
Necessary condition for unique decoding:

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

The Majority Rule Berlekamp/Massey algorithm



The Majority Rule Berlekamp/Massey algorithm



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

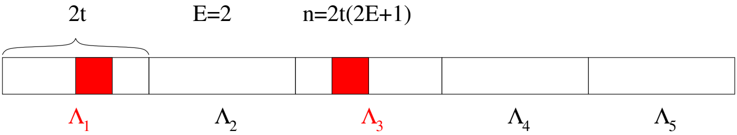
Output: $\Lambda(z)$ and $(a_0, \ldots, a_{n-1})$.

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

Output: $\Lambda(z)$ 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 $\Lambda(z)$;
- Use a *clean* segment to *clean-up* the sequence ;
- **return** $\Lambda(z)$ and $(a_0, a_1, \ldots)$;

Algorithm `SequenceCleanUp`

Input: $\Lambda(z) = z^t + \sum_{i=0}^{t-1} \lambda_i x^i$ where $\Lambda(0) \neq 0$
Input: $(a_0, \ldots, a_{n-1})$, where $n \geq t + 1$
Input: E, the maximum number of corrections to make
Input: k, such that (a_k, a_{k+2t-1}) is clean
Output: $(b_0, \ldots, b_{n-1})$ generated by Λ at distance $\leq E$ to $(a_0, \ldots, a_{n-1})$

Algorithm SequenceCleanUp

Input: $\Lambda(z) = z^t + \sum_{i=0}^{t-1} \lambda_i x^i$ where $\Lambda(0) \neq 0$
Input: $(a_0, \ldots, a_{n-1})$, where $n \geq t + 1$
Input: E, the maximum number of corrections to make
Input: k, such that (a_k, a_{k+2t-1}) is clean
Output: $(b_0, \ldots, b_{n-1})$ generated by Λ at distance $\leq E$ to $(a_0, \ldots, a_{n-1})$

begin
  $(b_0, \ldots, b_{n-1}) \leftarrow (a_0, \ldots, a_{n-1})$; $e, j \leftarrow 0$;
  $i \leftarrow k + 2t$;
  while $i \leq n - 1$ *and* $e \leq E$ **do**
    if Λ *does not satisfy* $(b_{i-t+1}, \ldots, b_i)$ **then**
      Fix b_i using $\Lambda(z)$ as a LFSR; $e \leftarrow e + 1$;
  return $(b_0, \ldots, b_{n-1}), e$

Algorithm SequenceCleanUp

Input: $\Lambda(z) = z^t + \sum_{i=0}^{t-1} \lambda_i x^i$ where $\Lambda(0) \neq 0$
Input: $(a_0, \ldots, a_{n-1})$, where $n \geq t+1$
Input: E, the maximum number of corrections to make
Input: k, such that (a_k, a_{k+2t-1}) is clean
Output: $(b_0, \ldots, b_{n-1})$ generated by Λ at distance $\leq E$ to $(a_0, \ldots, a_{n-1})$

begin
  $(b_0, \ldots, b_{n-1}) \leftarrow (a_0, \ldots, a_{n-1})$; $e, j \leftarrow 0$;
  $i \leftarrow k + 2t$;
  while $i \leq n-1$ *and* $e \leq E$ **do**
    if Λ *does not satisfy* $(b_{i-t+1}, \ldots, b_i)$ **then**
      Fix b_i using $\Lambda(z)$ as a LFSR; $e \leftarrow e + 1$;
  $i \leftarrow k - 1$;
  while $i \geq 0$ *and* $e \leq E$ **do**
    if Λ *does not satisfy* $(b_i, \ldots, b_{i+t-1})$ **then**
      Fix b_i using $z^t\Lambda(1/z)$ as a LFSR; $e \leftarrow e + 1$;
  return $(b_0, \ldots, b_{n-1}), e$

Algorithm SequenceCleanUp

Input: $\Lambda(z) = z^t + \sum_{i=0}^{t-1} \lambda_i x^i$ where $\Lambda(0) \neq 0$
Input: $(a_0, \ldots, a_{n-1})$, where $n \geq t+1$
Input: E, the maximum number of corrections to make
Input: k, such that (a_k, a_{k+2t-1}) is clean
Output: $(b_0, \ldots, b_{n-1})$ generated by Λ at distance $\leq E$ to $(a_0, \ldots, a_{n-1})$

begin
  $(b_0, \ldots, b_{n-1}) \leftarrow (a_0, \ldots, a_{n-1})$; $e, j \leftarrow 0$;
  $i \leftarrow k + 2t$;
  while $i \leq n-1$ *and* $e \leq E$ **do**
    if Λ *does not satisfy* $(b_{i-t+1}, \ldots, b_i)$ **then**
      Fix b_i using $\Lambda(z)$ as a LFSR; $e \leftarrow e+1$;
  $i \leftarrow k - 1$;
  while $i \geq 0$ *and* $e \leq E$ **do**
    if Λ *does not satisfy* $(b_i, \ldots, b_{i+t-1})$ **then**
      Fix b_i using $z^t\Lambda(1/z)$ as a LFSR; $e \leftarrow e+1$;
  return $(b_0, \ldots, b_{n-1}), e$

Finding a clean segment: case $E = 1$

$\Rightarrow$ only one error

$$(a_0, \ldots, a_{k-2}, b_{k-1} \neq a_{k-1}, a_k, a_{k+1}, a_{2t-1})$$

will be identified by the majority vote (2-to-1 majority).

Finding a clean segment: case $E \geq 2$

Multiple errors on one segment can still be generated by $\Lambda(z)$
$\Rightarrow$deceptive segments: not good for `SequenceCleanUp`

Example

$E = 3$: $(0, 1, 0, 2, 0, 4, 0, 8, \dots)$ $\Rightarrow \Lambda(z) = z^2 - 2$

Finding a clean segment: case $E \geq 2$

Multiple errors on one segment can still be generated by $\Lambda(z)$
⇒deceptive segments: not good for `SequenceCleanUp`

Example

$E = 3$: $(0, 1, 0, 2, 0, 4, 0, 8, \dots)$ ⇒$\Lambda(z) = z^2 - 2$

$$(\mathbf{1}, 1, \mathbf{2}, 2, \mathbf{4}, 4, 0, 8, 0, 16, 0, 32, \dots)$$

Finding a clean segment: case $E \geq 2$

Multiple errors on one segment can still be generated by $\Lambda(z)$
$\Rightarrow$deceptive segments: not good for `SequenceCleanUp`

Example

$E = 3$: $(0, 1, 0, 2, 0, 4, 0, 8, \dots)$ $\Rightarrow \Lambda(z) = z^2 - 2$

$$(\underbrace{\mathbf{1}, 1, \mathbf{2}, 2}_{z^2-2}, \underbrace{\mathbf{4}, 4, 0, 8}_{z^2+2z-2}, \underbrace{0, 16, 0, 32}_{z^2-2}, \dots)$$

Finding a clean segment: case $E \geq 2$

Multiple errors on one segment can still be generated by $\Lambda(z)$
$\Rightarrow$deceptive segments: not good for `SequenceCleanUp`

Example

$E = 3$: $(0, 1, 0, 2, 0, 4, 0, 8, \dots)$ $\Rightarrow \Lambda(z) = z^2 - 2$

$$(\underbrace{\mathbf{1}, 1, \mathbf{2}, 2}_{z^2-2}, \underbrace{\mathbf{4}, 4, 0, 8}_{z^2+2z-2}, \underbrace{0, 16, 0, 32}_{z^2-2}, \dots)$$

$(1, 1, 2, 2)$ is deceptive. Applying `SequenceCleanUp` with this clean segment produces

$$(\mathbf{1}, 1, \mathbf{2}, 2, \mathbf{4}, 4, 8, 8, 16, 16, 32, 32, 64, \dots)$$

Finding a clean segment: case $E \geq 2$

Multiple errors on one segment can still be generated by $\Lambda(z)$
$\Rightarrow$deceptive segments: not good for `SequenceCleanUp`

Example

$E = 3$: $(0, 1, 0, 2, 0, 4, 0, 8, \dots)$ $\Rightarrow \Lambda(z) = z^2 - 2$

$$(\underbrace{\mathbf{1}, 1, \mathbf{2}, 2}_{z^2-2}, \underbrace{\mathbf{4}, 4, 0, 8}_{z^2+2z-2}, \underbrace{0, 16, 0, 32}_{z^2-2}, \dots)$$

$(1, 1, 2, 2)$ is deceptive. Applying `SequenceCleanUp` with this clean segment produces

$$(\mathbf{1}, 1, \mathbf{2}, 2, \mathbf{4}, 4, 8, 8, 16, 16, 32, 32, 64, \dots)$$

$E > 3$? contradiction. Try $(0, 16, 0, 32)$ as a clean segment instead.

Success of the sequence clean-up

Theorem

If $n \geq t(2E+1)$, then a deceptive segment will necessarily be exposed by a failure of the condition $e \leq E$ in algorithm `SequenceCleanUp`*.*

Success of the sequence clean-up

Theorem

If $n \geq t(2E+1)$, then a deceptive segment will necessarily be exposed by a failure of the condition $e \leq E$ in algorithm `SequenceCleanUp`.

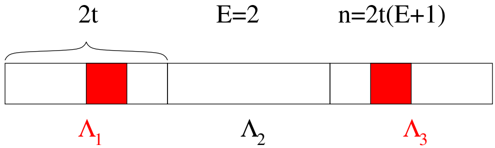
Corollary

$n \geq 2t(2E+1)$ is a necessary and sufficient condition for unique decoding of Λ and the corresponding sequence.

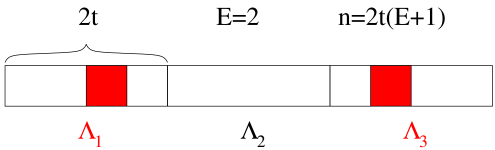
Remark

Also works with an upper bound $t \leq T$ on $\deg \Lambda$.

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



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



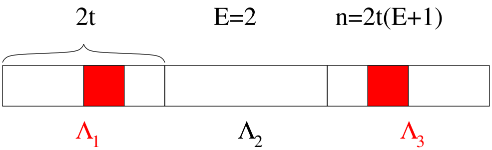
Input: $(a_0, \ldots, a_{n-1}) + \varepsilon$, where $n = 2t(E+1)$, $weight(\varepsilon) \leq E$, and $(a_0, \ldots, a_{n-1})$ minimally generated by Λ of degree t, where $\Lambda(0) \neq 0$.

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 $(\Lambda_i(z), (a_0^{(i)}, \ldots, a_{n-1}^{(i)}))_i$;

Properties

- The list contains the right solution $(\Lambda, (a_0, \ldots, a_{n-1}))$

Properties

- The list contains the right solution $(\Lambda, (a_0, \ldots, a_{n-1}))$
- $n \geq 2t(E+1)$ is the tightest bound to enable syndrome decoding (BMA on a clean sequence of length $2t$).

Example

$n = 2t(E+1) - 1$ and $\varepsilon = (\underbrace{0, \ldots, 0}_{2t-1}, 1, \underbrace{0, \ldots, 0}_{2t-1}, 1 \ldots, 1, \underbrace{0, \ldots, 0}_{2t-1})$.

Then $(a_0, \ldots, a_{n-1}) + \varepsilon$ has no length $2t$ clean segment.

Outline

Berlekamp/Massey algorithm with errors

Bounds on the decoding capacity

Decoding algorithms

Sparse Polynomial Interpolation with errors

Relations to Reed-Solomon decoding

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/Tiwari 1988:

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

$\Rightarrow$only need $2t$ entries to find $\Lambda(\lambda)$ (using BMA)

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/Tiwari 1988:

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

$\Rightarrow$only need $2t$ entries to find $\Lambda(\lambda)$ (using BMA)

$\Rightarrow$only need $2T(2E+1)$ with $e \leq E$ errors and $t \leq T$.

Ben-Or & Tiwari's Algorithm

Input: $(a_0, \ldots, a_{2t-1})$ where $a_i = f(p^i)$
Input: t, the numvber of (non-zero) terms of $f(x) = \sum_{j=1}^{t} c_j x^{e_j}$
Output: $f(x)$

begin
- Run BMA on $(a_0, \ldots, a_{2t-1})$ to find $\Lambda(z)$
- Find roots of $\Lambda(z)$ (polynomial factorization)
- Recover e_j by repeated division (by p)
- Recover c_j by solving the transposed Vandermonde system

$$\begin{bmatrix} (p^0)^{e_1} & (p^0)^{e_2} & \ldots & (p^0)^{e_t} \\ (p^1)^{e_1} & (p^1)^{e_2} & \ldots & (p^1)^{e_t} \\ \vdots & \vdots & & \vdots \\ (p^t)^{e_1} & (p^t)^{e_2} & \ldots & (p^t)^{e_t} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_t \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{t-1} \end{bmatrix}$$

Outline

Berlekamp/Massey algorithm with errors

- Bounds on the decoding capacity
- Decoding algorithms

Sparse Polynomial Interpolation with errors

Relations to Reed-Solomon decoding

Blahut's theorem

Theorem (Blahut)

The D.F.T of a vector of weight t has linear complexity at most t

- $\mathsf{DFT}_\omega(v) \Leftrightarrow \mathsf{Vandemonde}(\omega^0, \omega^1, \omega^2, \ldots)v \Leftrightarrow Eval_{\omega^0,\omega^1,\omega^2,\ldots}(v)$

Blahut's theorem

Theorem (Blahut)

The D.F.T of a vector of weight t has linear complexity at most t

- $\text{DFT}_\omega(v) \Leftrightarrow \text{Vandemonde}(\omega^0, \omega^1, \omega^2, \dots)v \Leftrightarrow Eval_{\omega^0,\omega^1,\omega^2,\dots}(v)$
- Univariate Ben-Or & Tiwari as a corollary

Blahut's theorem

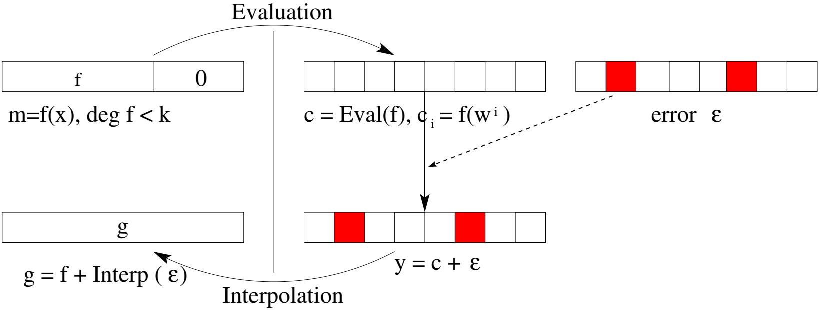
Theorem (Blahut)

The D.F.T of a vector of weight t has linear complexity at most t

- $\text{DFT}_\omega(v) \Leftrightarrow \text{Vandemonde}(\omega^0, \omega^1, \omega^2, \dots)v \Leftrightarrow Eval_{\omega^0,\omega^1,\omega^2,\dots}(v)$
- Univariate Ben-Or & Tiwari as a corollary
- Reed-Solomon codes: evaluation of a sparse error $\Rightarrow$BMA

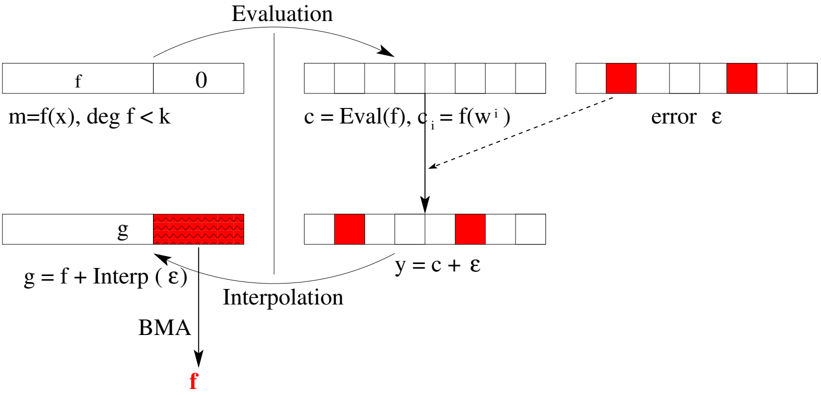
Reed-Solomon codes as Evaluation codes

$$\mathcal{C} = \{(f(\omega^1), \ldots, f(\omega^n)) \mid \deg f < k\}$$



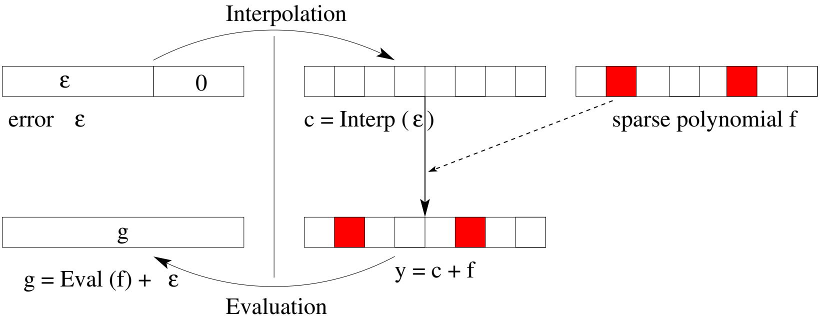
Reed-Solomon codes as Evaluation codes

$$\mathcal{C} = \{(f(\omega^1), \ldots, f(\omega^n)) | \deg f < k\}$$



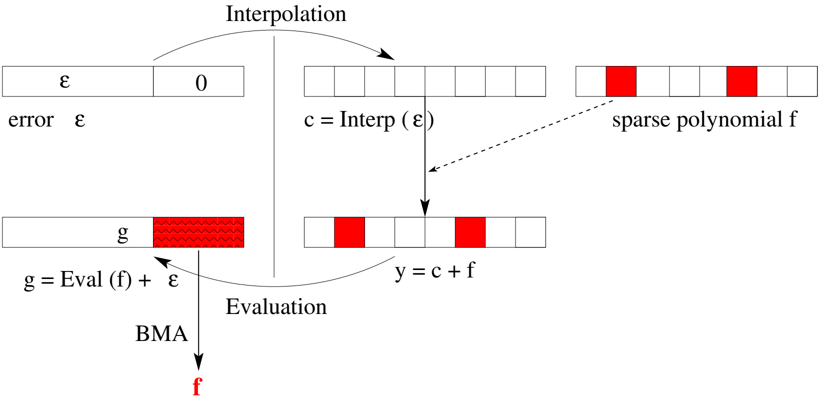
Sparse interpolation with errors

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



Sparse interpolation with errors

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



Same problems?

Interchanging **Evaluation** and **Interpolation**

Let $V_\omega = \text{Vandermonde}(\omega, \omega^2, \ldots, \omega^n)$. Then $(V_\omega)^{-1} = \frac{1}{n} V_{\omega^{-1}}$

Given g, find f, t-sparse and an error ε such that

$$\begin{aligned} g &= V_\omega f + \varepsilon \\ V_{\omega^{-1}} g &= nf + V_{\omega^{-1}} \epsilon \end{aligned}$$

Same problems?

Interchanging Evaluation and Interpolation

Let $V_\omega = \text{Vandermonde}(\omega, \omega^2, \ldots, \omega^n)$. Then $(V_\omega)^{-1} = \frac{1}{n} V_{\omega^{-1}}$

Given g, find f, t-sparse and an error ε such that

$$\begin{aligned} g &= V_\omega f + \varepsilon \\ V_{\omega^{-1}} g &= \underbrace{nf}_{\text{weight t error}} + \underbrace{V_{\omega^{-1}} \epsilon}_{\text{RS code word}} \end{aligned}$$

Reed-Solomon decoding: unique solution provided ε has $2t$ consecutive trailing 0's

$\Leftrightarrow$ clean segment of length $2t$

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

Same problems?

Interchanging Evaluation and Interpolation

Let $V_\omega = \text{Vandermonde}(\omega, \omega^2, \ldots, \omega^n)$. Then $(V_\omega)^{-1} = \frac{1}{n} V_{\omega^{-1}}$

Given g, find f, t-sparse and an error ε such that

$$\begin{aligned} g &= V_\omega f + \varepsilon \\ V_{\omega^{-1}} g &= \underbrace{nf}_{\text{weight t error}} + \underbrace{V_{\omega^{-1}} \epsilon}_{\text{RS code word}} \end{aligned}$$

Reed-Solomon decoding: unique solution provided ε has $2t$ consecutive trailing 0's

$\Leftrightarrow$ clean segment of length $2t$

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

BUT: location of the syndrome, is a priori unknown

$\Rightarrow$ no uniqueness

Applications and Perspectives

Sparse interpolation with noise and outliers

[Giesbrecht, Labahn &Lee'06] [Kaltofen, Lee, Yang'11]:

Termination criteria for BMA:

Exact singularity $\leftrightarrow$ illconditionnedness

Now combined with outliers

Applications and Perspectives

Sparse interpolation with noise and outliers

[Giesbrecht, Labahn &Lee'06] [Kaltofen, Lee, Yang'11]:

Termination criteria for BMA:

Exact singularity $\leftrightarrow$ illconditionnedness

Now combined with outliers

Perspectives:

- surprising impact of noise on the sparsity: does not degenerate to dense

Applications and Perspectives

Sparse interpolation with noise and outliers

[Giesbrecht, Labahn &Lee'06] [Kaltofen, Lee, Yang'11]:

Termination criteria for BMA:

Exact singularity $\leftrightarrow$ illconditionnedness

Now combined with outliers

Perspectives:

- surprising impact of noise on the sparsity: does not degenerate to dense
- Sparse rational function reconstruction with errors:

  dense case: Berlekamp/Welsh decoding and Padé approximant fit well together.

  sparse case ?

Applications and Perspectives

Sparse interpolation with noise and outliers

[Giesbrecht, Labahn &Lee'06] [Kaltofen, Lee, Yang'11]:

Termination criteria for BMA:

Exact singularity $\leftrightarrow$ illconditionnedness

Now combined with outliers

Perspectives:

- surprising impact of noise on the sparsity: does not degenerate to dense
- Sparse rational function reconstruction with errors:

  dense case: Berlekamp/Welsh decoding and Padé approximant fit well together.

  sparse case ?
- application to k-error linear complexity (symmetric crypto)
- ...