An extension of the Error Correcting Pairs algorithm

Alain Couvreur Isabella Panaccione

Inria, LIX

Codes, Cryptology and Curves
Celebrating the influence of Ruud Pellikaan

08/03/2019

Error Correcting Pairs algorithm

PECP for Reed-Solomon codes

PECP for Algebraic Geometry codes

Algorithms for Reed Solomon codes

$t \leq \left\lfloor \frac{d-1}{2} \right\rfloor$ Berlekamp-Welch [1]

[1] L. R. Welch, E.R.Berlekamp. Error Correction for Algebraic Block Codes. United States Patent, 1986.

Algorithms for Reed Solomon codes

$t \leq \left\lfloor \frac{d-1}{2} \right\rfloor$ Berlekamp-Welch [1] Error Correcting Pairs [2]

[2] R. Pellikaan. On decoding by error location and dependent sets of error positions. Discrete Mathematics, 1992.

Algorithms for Reed Solomon codes

$t \leq \left\lfloor \frac{d-1}{2} \right\rfloor$	Berlekamp-Welch [1]	Error Correcting Pairs [2]
	$\downarrow$	
$t > \left\lfloor \frac{d-1}{2} \right\rfloor$	Sudan [3]	

[3] M. Sudan. Decoding of Reed-Solomon codes beyond the error-correction bound. Journal of Complexity, 1997.

Algorithms for Reed Solomon codes

$t \leq \left\lfloor \frac{d-1}{2} \right\rfloor$	Berlekamp-Welch [1]	Error Correcting Pairs [2]
	$\downarrow$	
$t > \left\lfloor \frac{d-1}{2} \right\rfloor$	Sudan [3] Power Decoding [4]	

[4] G. Schmidt, V. R. Sidorenko, M. Bossert. Syndrome Decoding of Reed-Solomon Codes Beyond Half of the Minimum Distance based on Shift-Register Synthesis. IEEE Transactions on Information Theory, 2010.

Algorithms for Reed Solomon codes

$t \leq \left\lfloor \frac{d-1}{2} \right\rfloor$	Berlekamp-Welch [1]	Error Correcting Pairs [2]
	$\downarrow$	$\downarrow$
$t > \left\lfloor \frac{d-1}{2} \right\rfloor$	Sudan [3] Power Decoding [4]	?

Error Correcting Pairs algorithm

Problem

Let $C \subseteq \mathbb{F}_q^n$ be a linear code and $y \in \mathbb{F}_q^n$. Given $t \in \mathbb{N}$, find a codeword c such that

$$\mathrm{d}(y, c) \leq t.$$

Problem

Let $C \subseteq \mathbb{F}_q^n$ be a linear code and $y \in \mathbb{F}_q^n$. Given $t \in \mathbb{N}$, find a codeword c such that

$$\mathrm{d}(y, c) \leq t.$$

Hypothesis

There exist $c \in C$ and $e = (e_1, \ldots, e_n) \in \mathbb{F}_q^n$ with $\mathrm{w}(e) = t$ such that

$$y = c + e.$$

We denote the support of the error vector by

$$I = \{i \in \{1, \ldots, n\} \mid e_i \neq 0\}.$$

Error Correcting Pairs algorithm:

- Localisation of errors: find J such that $I \subseteq J$;

Error Correcting Pairs algorithm:

- Localisation of errors: find J such that $I \subseteq J$;
- Syndromes linear system: recover e.

Error Correcting Pairs algorithm:

- Localisation of errors: find J such that $I \subseteq J$;
- Syndromes linear system: recover e.

Error Correcting Pairs (ECP)

Given a linear code $C \subseteq \mathbb{F}_q^n$, a pair of linear codes (A, B) with $A, B \subseteq \mathbb{F}_q^n$ is a t-error correcting pair for C if

- $A * B \subseteq C^\perp$;
- $\dim(A) > t$;
- $\mathrm{d}(B^\perp) > t$;
- $\mathrm{d}(A) + \mathrm{d}(C) > n$.

Theorem (R. Pellikaan, 1992)

Let $C \subseteq \mathbb{F}_q^n$ be a linear code. If there exists a t-error correcting pair for C, then for all $y \in \mathbb{F}_q^n$ such that

$$y = c + e,$$

with $c \in C$ and $\mathrm{w}(e) \leq t$, the ECP algorithm recovers c with complexity $O(n^3)$.

Theorem (R. Pellikaan, 1992)

Let $C \subseteq \mathbb{F}_q^n$ be a linear code. If there exists a t-error correcting pair for C, then for all $y \in \mathbb{F}_q^n$ such that

$$y = c + e,$$

with $c \in C$ and $\mathrm{w}(e) \leq t$, the ECP algorithm recovers c with complexity $O(n^3)$.

Proposition

If a linear code C has a t-error correcting pair, then

$$t \leq \left\lfloor \frac{\mathrm{d}(C) - 1}{2} \right\rfloor.$$

Let $J = \{j_1, \ldots, j_s\} \subset \{1, \ldots, n\}$ and $x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$.
We denote

- $x_J := (x_{j_1}, \ldots, x_{j_s})$ (puncturing);
- $Z(x) := \{i \in \{1, \ldots, n\} \mid x_i = 0\}$.

Let $J = \{j_1, \ldots, j_s\} \subset \{1, \ldots, n\}$ and $x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$.
We denote

- $x_J := (x_{j_1}, \ldots, x_{j_s})$ (puncturing);
- $Z(x) := \{i \in \{1, \ldots, n\} \mid x_i = 0\}$.

Moreover, if $A \subseteq \mathbb{F}_q^n$ we will indicate

- $A_J := \{a_J \mid a \in A\} \subseteq \mathbb{F}_q^{|J|}$;
- $Z(A) := \{i \in \{1, \ldots, n\} \mid a_i = 0 \;\; \forall a \in A\}$;

Let $J = \{j_1, \ldots, j_s\} \subset \{1, \ldots, n\}$ and $x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$.
We denote

- $x_J := (x_{j_1}, \ldots, x_{j_s})$ (puncturing);
- $Z(x) := \{i \in \{1, \ldots, n\} \mid x_i = 0\}$.

Moreover, if $A \subseteq \mathbb{F}_q^n$ we will indicate

- $A_J := \{a_J \mid a \in A\} \subseteq \mathbb{F}_q^{|J|}$;
- $Z(A) := \{i \in \{1, \ldots, n\} \mid a_i = 0 \ \ \forall a \in A\}$;
- $A(J) := \{a \in A \mid a_J = 0\} \subseteq \mathbb{F}_q^n$ (shortening).

Localisation of errors

We define $M \coloneqq \{a \in A \mid \langle a * y, b \rangle = 0 \ \ \forall b \in B\}$.

Lemma

Let y, $I = \text{supp}(e)$ and M as above. If $A * B \subseteq C^{\perp}$, then

- $A(I) \subseteq M \subseteq A$;

Localisation of errors

We define $M \coloneqq \{a \in A \mid \langle a * y, b \rangle = 0 \;\; \forall b \in B\}$.

Lemma

Let y, $I = \mathrm{supp}(e)$ and M as above. If $A * B \subseteq C^{\perp}$, then

- $A(I) \subseteq M \subseteq A$;
- if $\mathrm{d}(B^{\perp}) > t$, then $A(I) = M$;

Localisation of errors

We define $M := \{a \in A \mid \langle a * y, b\rangle = 0 \;\; \forall b \in B\}$.

Lemma

Let y, $I = \mathrm{supp}(e)$ and M as above. If $A * B \subseteq C^\perp$, then

- $A(I) \subseteq M \subseteq A$;
- if $\mathrm{d}(B^\perp) > t$, then $A(I) = M$;
- if $\dim(A) > t$, then $A(I) \neq 0$.

Localisation of errors

We define $M := \{a \in A \mid \langle a * y, b\rangle = 0 \ \ \forall b \in B\}$.

Lemma

Let y, $I = \mathrm{supp}(e)$ and M as above. If $A * B \subseteq C^\perp$, then

- $A(I) \subseteq M \subseteq A$;
- if $\mathrm{d}(B^\perp) > t$, then $A(I) = M$;
- if $\dim(A) > t$, then $A(I) \neq 0$.

Proof of $A(I) \subseteq M$: given $a \in A(I)$, we get for all $b \in B$

$$\langle a * y, b\rangle = \underbrace{\langle a * c, b\rangle}_{\langle a*b, c\rangle} + \underbrace{\langle a * e, b\rangle}_{\langle 0, b\rangle} = 0.$$

Localisation of errors

We define $M := \{a \in A \mid \langle a * y, b\rangle = 0 \;\; \forall b \in B\}$.

Lemma

Let y, $I = \mathrm{supp}(e)$ and M as above. If $A * B \subseteq C^{\perp}$, then

- $A(I) \subseteq M \subseteq A$;
- if $\mathrm{d}(B^{\perp}) > t$, then $A(I) = M$;
- if $\dim(A) > t$, then $A(I) \neq 0$.

Proof of $A(I) \subseteq M$: given $a \in A(I)$, we get for all $b \in B$

$$\langle a * y, b\rangle = \underbrace{\langle a * c, b\rangle}_{\langle a*b, c\rangle} + \underbrace{\langle a * e, b\rangle}_{\langle 0, b\rangle} = 0.$$

$\longrightarrow$ we take $J := Z(M)$.

Recovering e

Let $H \in \mathcal{M}(n, m)$, and H^i its columns. Given $J \subseteq \{1, \ldots, m\}$, we define

$$H_J = (H^j)^{j \in J}.$$

Let us consider a full rank parity check matrix H for C.

Lemma

If $\mathrm{d}(A) + \mathrm{d}(C) > n$ and $I \subseteq J$, then there exists an unique solution for the system

$$H_J \cdot E^T = H \cdot y^T.$$

Recovering e

Let $H \in \mathcal{M}(n, m)$, and H^i its columns. Given $J \subseteq \{1, \dots, m\}$, we define

$$H_J = (H^j)^{j \in J}.$$

Let us consider a full rank parity check matrix H for C.

Lemma

If $\mathrm{d}(A) + \mathrm{d}(C) > n$ and $I \subseteq J$, then there exists an unique solution for the system

$$H_J \cdot E^T = H \cdot y^T.$$

$\longrightarrow$ we recover e.

PECP for Reed-Solomon codes

Let $C \subseteq \mathbb{F}_q^n$ be a RS[n,k] code. There exists $f \in \mathbb{F}_q[x]_{<k}$ such that $c = (f(x_1), \ldots, f(x_n))$. Let us take

$$A = RS[n, t+1], \qquad B^{\perp} = RS[n, t+k].$$

Let $C \subseteq \mathbb{F}_q^n$ be a RS[n,k] code. There exists $f \in \mathbb{F}_q[x]_{<k}$ such that $c = (f(x_1), \dots, f(x_n))$. Let us take

$$A = RS[n, t+1], \qquad B^\perp = RS[n, t+k].$$

$\dim(A) > t$
$A * B \subseteq C^\perp$
$\mathrm{d}(A) + \mathrm{d}(C) > n$

Let $C \subseteq \mathbb{F}_q^n$ be a RS[n,k] code. There exists $f \in \mathbb{F}_q[x]_{<k}$ such that $c = (f(x_1), \ldots, f(x_n))$. Let us take

$$A = RS[n, t+1], \qquad B^{\perp} = RS[n, t+k].$$

$\dim(A) > t$ obvious

$A * B \subseteq C^{\perp}$

$\mathrm{d}(A) + \mathrm{d}(C) > n$

Let $C \subseteq \mathbb{F}_q^n$ be a RS[n,k] code. There exists $f \in \mathbb{F}_q[x]_{<k}$ such that $c = (f(x_1), \dots, f(x_n))$. Let us take

$$A = RS[n, t+1], \qquad B^{\perp} = RS[n, t+k].$$

$\dim(A) > t$	obvious
$A * B \subseteq C^{\perp}$	$A * C = B^{\perp}$
$\mathrm{d}(A) + \mathrm{d}(C) > n$	

Let $C \subseteq \mathbb{F}_q^n$ be a RS[n,k] code. There exists $f \in \mathbb{F}_q[x]_{<k}$ such that $c = (f(x_1), \dots, f(x_n))$. Let us take

$$A = RS[n, t+1], \qquad B^\perp = RS[n, t+k].$$

$\dim(A) > t$	obvious
$A * B \subseteq C^\perp$	$A * C = B^\perp$
$\mathrm{d}(A) + \mathrm{d}(C) > n$	$t < d$

Let $C \subseteq \mathbb{F}_q^n$ be a RS[n,k] code. There exists $f \in \mathbb{F}_q[x]_{<k}$ such that $c = (f(x_1), \dots, f(x_n))$. Let us take

$$A = RS[n, t+1], \qquad B^\perp = RS[n, t+k].$$

$\dim(A) > t$	obvious
$A * B \subseteq C^\perp$	$A * C = B^\perp$
$\mathrm{d}(A) + \mathrm{d}(C) > n$	$t < d$

Proposition

We have that $\mathrm{d}(B^\perp) > t$ if and only if

$$t \leq \left\lfloor \frac{\mathrm{d}(C) - 1}{2} \right\rfloor.$$

Berlekamp-Welch key equations and the choice of M

Berlekamp Welch algorithm's key equation (Roth)

Let $\Lambda(x) \coloneqq \prod_{i \in I}(x - x_i)$ and $N(x) \coloneqq \Lambda(x) f(x)$. Then

$$(\Lambda(x_i))_i * y = (N(x_i))_i.$$

We get

- $(N(x_1), \ldots, N(x_n)) \in B^{\perp} = RS[t+k]$;
- $(\Lambda(x_1), \ldots, \Lambda(x_n)) \in A(I) = RS[t+1](I)$;

Berlekamp-Welch key equations and the choice of M

Berlekamp Welch algorithm's key equation (Roth)

Let $\Lambda(x) := \prod_{i \in I}(x - x_i)$ and $N(x) := \Lambda(x) f(x)$. Then

$$(\Lambda(x_i))_i * y = (N(x_i))_i.$$

We get

- $(N(x_1), \ldots, N(x_n)) \in B^\perp = RS[t+k]$;
- $(\Lambda(x_1), \ldots, \Lambda(x_n)) \in A(I) = RS[t+1](I)$;
- $(\Lambda(x_1), \ldots, \Lambda(x_n)) \in \underbrace{\{a \in A \mid \langle a * y, b \rangle = 0 \ \ \forall b \in B\}}_{M}$.

Algorithms for Reed Solomon codes

$t \leq \left\lfloor \frac{d-1}{2} \right\rfloor$	Berlekamp-Welch	Error Correcting Pairs
	$\downarrow$	$\downarrow$
$t > \left\lfloor \frac{d-1}{2} \right\rfloor$	Sudan Power Decoding	?

Algorithms for Reed Solomon codes

$t \leq \left\lfloor \frac{d-1}{2} \right\rfloor$	Berlekamp-Welch	Error Correcting Pairs
	$\downarrow$	$\downarrow$
$t > \left\lfloor \frac{d-1}{2} \right\rfloor$	Sudan Power Decoding	?

Proposition

We have that $\mathrm{d}(B^{\perp}) > t$ if and only if

$$t \leq \left\lfloor \frac{\mathrm{d}(C) - 1}{2} \right\rfloor.$$

Power Error Correcting Pairs algorithm with power $\ell = 2$

Error Locating Pair

Given A, B, C linear codes of length n, (A, B) is a t-error locating pair (QECP) for C if

- $A * B \subseteq C^{\perp}$;
- $\dim(A) > t$;
- $\mathrm{d}(A) + \mathrm{d}(C) > n$.

Pellikaan, 1992:

> *If I is an independent t-set of error positions with respect to B, where (A, B) is a t-error locating pair for C, then the algorithm corrects any word with error supported at I.*

Power Error Correcting Pairs algorithm with power $\ell = 2$

Error Locating Pair

Given A, B, C linear codes of length n, (A, B) is a t-error locating pair (QECP) for C if

- $A * B \subseteq C^{\perp}$;
- $\dim(A) > t$;
- $\mathrm{d}(A) + \mathrm{d}(C) > n$.

Pellikaan, 1992:

> *If I is an independent t-set of error positions with respect to B, where (A, B) is a t-error locating pair for C, then the algorithm corrects any word with error supported at I.*

Before we used "If $A * B \subseteq C^{\perp}$ and $\mathrm{d}(B^{\perp}) > t$, then $A(I) = M$."

Let us define e' this way

$$y^{*2} = c^{*2} + \underbrace{2c * e + e^{*2}}_{e'}.$$

Lemma

We get $\mathrm{supp}(e') \subseteq I = \mathrm{supp}(e)$.

Let us define e' this way

$$y^{*2} = c^{*2} + \underbrace{2c * e + e^{*2}}_{e'}.$$

Lemma

We get $\mathrm{supp}(e') \subseteq I = \mathrm{supp}(e)$.

Power Decoding algorithm's key equations

If $\Lambda(x) = \prod_{i \in I}(x - x_i)$ and $N_j(x) := \Lambda(x) f^j(x)$ for $j = 1, 2$, then

$$\begin{cases} (\Lambda(x_i))_i * y = (N_1(x_i))_i \\ (\Lambda(x_i))_i * y^{*2} = (N_2(x_i))_i \end{cases}.$$

Hence, if we consider $A = RS[n, t+1]$, $B^{\perp} = RS[n, t+k]$ as before, we get

- $(N_1(x_1), \ldots, N_1(x_n)) \in B^{\perp}$;
- $(N_2(x_1), \ldots, N_2(x_n)) \in B^{\perp} * C$;

Hence, if we consider $A = RS[n, t+1]$, $B^{\perp} = RS[n, t+k]$ as before, we get

- $(N_1(x_1), \ldots, N_1(x_n)) \in B^{\perp}$;
- $(N_2(x_1), \ldots, N_2(x_n)) \in B^{\perp} * C$;
- $(\Lambda(x_1), \ldots, \Lambda(x_n)) \in A(I),\ M_1 \cap M_2$.

where M_1 and M_2 are defined this way

$$
\begin{aligned}
M_1 &:= \{a \in A \mid \langle a * y, b \rangle = 0 \ \ \forall b \in B\}, \\
M_2 &:= \{a \in A \mid \langle a * y^{*2}, v \rangle = 0 \ \ \forall v \in (B^{\perp} * C)^{\perp}\}.
\end{aligned}
$$

Hence, if we consider $A = RS[n, t+1]$, $B^{\perp} = RS[n, t+k]$ as before, we get

- $(N_1(x_1), \dots, N_1(x_n)) \in B^{\perp}$;
- $(N_2(x_1), \dots, N_2(x_n)) \in B^{\perp} * C$;
- $(\Lambda(x_1), \dots, \Lambda(x_n)) \in A(I),\ \ M_1 \cap M_2$.

where M_1 and M_2 are defined this way

$$\begin{aligned} M_1 :=& \{a \in A \mid \langle a * y, b\rangle = 0 \ \ \forall b \in B\}, \\ M_2 :=& \{a \in A \mid \langle a * y^{*2}, v\rangle = 0 \ \ \forall v \in (B^{\perp} * C)^{\perp}\}. \end{aligned}$$

$\longrightarrow$ we take $M = M_1 \cap M_2$.

PECP algorithm:

- compute $M = M_1 \cap M_2$ (linear system);
- compute $J = Z(M)$;
- solve the syndrom linear system.

This algorithm can be run on all codes with an ELP.

PECP algorithm:

- compute $M = M_1 \cap M_2$ (linear system);
- compute $J = Z(M)$;
- solve the syndrom linear system.

This algorithm can be run on all codes with an ELP.

Lemma

If $A * B \subseteq C^{\perp}$, then $A(I) \subseteq M = M_1 \cap M_2 \subseteq A$.

We look for a **necessary condition** to have $M = A(I)$.

PECP algorithm:

- compute $M = M_1 \cap M_2$ (linear system);
- compute $J = Z(M)$;
- solve the syndrom linear system.

This algorithm can be run on all codes with an ELP.

Lemma

If $A * B \subseteq C^\perp$, then $A(I) \subseteq M = M_1 \cap M_2 \subseteq A$.

We look for a **necessary condition** to have $M = A(I)$.

Since $M(I) = A(I)$, we get the implications:

$$M = A(I) \iff M(I) = M \iff M_I = \{0\}.$$

Given $a \in A$, we have by definition of M_1

$$a \in M_1 \iff \langle a * y, b \rangle = 0 \quad \forall b \in B.$$

If $A * B \subseteq C^\perp$, this is equivalent to $a_I \in (e * B)_I^\perp$.

Given $a \in A$, we have by definition of M_1

$$a \in M_1 \iff \langle a * y, b \rangle = 0 \ \ \forall b \in B.$$

If $A * B \subseteq C^\perp$, this is equivalent to $a_I \in (e * B)_I^\perp$.

In the same way, given $a \in A$, it holds

$$a \in M_2 \iff a_I \in (e' * (B^\perp * C)^\perp)_I^\perp.$$

Lemma

We have $(M_1 \cap M_2)_I = (e * B)_I^\perp \cap (e' * (B^\perp * C)^\perp)_I^\perp \cap A_I$.

Remark

Since $A = RS[n, t+1]$ is MDS, then $A_I = \mathbb{F}_q^t$.

Hence $(M_1 \cap M_2)_I = (e * B)_I^{\perp} \cap (e' * (B^{\perp} * C)^{\perp})_I^{\perp}$.

Remark

Since $A = RS[n, t+1]$ is MDS, then $A_I = \mathbb{F}_q^t$.

Hence $(M_1 \cap M_2)_I = (e * B)_I^\perp \cap (e' * (B^\perp * C)^\perp)_I^\perp$.

A necessary condition for $(M_1 \cap M_2)_I$ to be the null space is

$$\dim((e * B)_I^\perp) + \dim((e' * (B^\perp * C)^\perp)_I^\perp) \leq t.$$

This inequality implies the following

Necessary condition

$$\dim(B) + \dim((B^\perp * C)^\perp) \geq t.$$

Decoding radius for Reed-Solomon codes and $\ell = 2$

We get, as for the Power Decoding algorithm with power 2,

$$t \leq \frac{2n - 3k + 1}{3}.$$

It is possible to write the algorithm for a general power ℓ.

Decoding radius for Reed-Solomon codes and $\ell = 2$

We get, as for the Power Decoding algorithm with power 2,

$$t \leq \frac{2n - 3k + 1}{3}.$$

It is possible to write the algorithm for a general power ℓ.

For **Reed-Solomon codes**, PECP has the same decoding radius as the Power Decoding algorithm, that is $t_{pow} = \frac{2n\ell - k\ell(\ell+1) + \ell(\ell-1)}{2(\ell+1)}$.

Complexity

PECP(ℓ):

(i) find $M = \bigcap_{i=1}^{\ell} M_i$;

(ii) given J, find c.

The main cost is the one of step (i), which reduces to a linear system of $O(n\ell)$ equations in

$$t + 1 = O\left(\frac{2n\ell + \ell(\ell+1) + 2}{2(\ell+1)}\right) = O(n)$$

unknowns. Hence we get the cost $O(n^3\ell)$.

PECP for Algebraic Geometry codes

Let χ be a smooth projective curve, $\mathcal{P} = \{P_1, \dots, P_n\} \subseteq \chi$, G a divisor for χ with $\mathrm{supp}(G) \cap \mathcal{P} = \emptyset$ and

$$C = C_L(\chi, \mathcal{P}, G).$$

Theorem

There exists a t-error locating pair for C such that the necessary condition gives the correcting radius

$$t \leq \underbrace{\frac{2n\ell - \ell(\ell+1)\deg(G) - 2\ell}{2(\ell+1)} - g}_{t_{basic}, t_{pow}[SW98]} + \frac{g}{\ell+1}.$$

Future tasks:

- study of the failure cases of the Power Decoding algorithm and the PECP algorithm for Reed-Solomon codes;
- examine the possibility to improve PECP algorithm's decoding radius for algebraic-geometry codes;
- is it possible to design a multiplicity version of ECP algorithm?

Thanks for your attention!