DECODING ALGORITHMS

Patrick Fitzpatrick

University College Cork
Ireland

S^3Cm, 2–11 July 2008

Outline

Suppose that the transmitted word is $c \in C$ and we have received a word $y \in \mathbb{F}_q^n$. The error in the transmission is $e = y - c$.

Suppose that the transmitted word is $c \in C$ and we have received a word $y \in \mathbb{F}_q^n$. The error in the transmission is $e = y - c$.

Syndrome

We call

$$s(y) = Hy^t \in F_q^{n-k}$$

the **syndrome** of y. Notice that, $c \in C$ if and only if $s(c) = 0$. Therefore, since the syndrome is a linear map, $s(y) = s(c + e) = s(c) + s(e) = s(e)$.

Suppose that the transmitted word is $c \in C$ and we have received a word $y \in \mathbb{F}_q^n$. The error in the transmission is $e = y - c$.

Syndrome

We call

$$s(y) = Hy^t \in F_q^{n-k}$$

the **syndrome** of y. Notice that, $c \in C$ if and only if $s(c) = 0$. Therefore, since the syndrome is a linear map, $s(y) = s(c+e) = s(c) + s(e) = s(e)$.

The syndrome of a received vector is the linear combination of the columns of H corresponding with the positions of the error weighted by the values of the errors.

Cosets and coset leaders

Consider in $\mathbb{F}_q^n$ the set of (group) cosets of C: $C = 0 + C, a_2 + C, \ldots, a_{q^{n-k}} + C$. Every element in a typical coset $a + C$ has the same syndrome $s(a)$. Suppose the received word y lies in $a + C$ so $y = a + c$ for some c. We could decode y to c by subtracting a from c. If we decode y to any other codeword c' this means decoding y as $y - (a + c) + c' = y - [a + (c' - c)] = y - [a + c'']$ i.e. subtracting another element of the same coset $a + C$.

Cosets and coset leaders

Consider in $\mathbb{F}_q^n$ the set of (group) cosets of C: $C = 0 + C, a_2 + C, \ldots, a_{q^{n-k}} + C$. Every element in a typical coset $a + C$ has the same syndrome $s(a)$. Suppose the received word y lies in $a + C$ so $y = a + c$ for some c. We could decode y to c by subtracting a from c. If we decode y to any other codeword c' this means decoding y as $y - (a + c) + c' = y - [a + (c' - c)] = y - [a + c'']$ i.e. subtracting another element of the same coset $a + C$.

This means we should choose an element $a + c''$ of smallest weight in the coset $a + C$ and always decode any y in that coset by subtracting this element.

Coset leaders

An element of minimum weight in a coset is called a **coset leader**. Coset leaders are not necessarily unique i.e. there may be more than one element of smallest weight in the coset.

Proposition

A coset of C has at most one element of weight $\leq t = \lfloor d - 1/2 \rfloor$.

Proof. If u, v lie in the same coset, and both have weight $\leq t$, then $u - v \in C$ and $wt(u - v) \leq wt(u) + wt(v) \leq 2t < d$. Therefore, $u = v$.

Proposition

A coset of C has at most one element of weight $\leq t = \lfloor d - 1/2 \rfloor$.
Proof. If u, v lie in the same coset, and both have weight $\leq t$, then $u - v \in C$ and $wt(u - v) \leq wt(u) + wt(v) \leq 2t < d$. Therefore, $u = v$.

Decoding is uniquely defined if and only if the coset of y has a unique leader. The proposition guarantees that if the number of errors is at most t then decoding is unique (and is maximum likelihood decoding).

Decoding using coset leaders

Algorithm

1. For each coset choose a coset leader as representative.
2. Construct a table matching syndromes to coset leaders.
3. If y is received then calculate $s(y)$.
4. Find the corresponding coset leader x.
5. Decode as $y - x$.

Note that this algorithm is only feasible for very small codes.

Decoding BCH Codes

Let C be a narrow sense ($b = 1$) BCH code length n and designed distance $d = 2t + 1$ over F_q, with generator polynomial g (we assume that q and n are relatively prime). Let α be a primitive $n\mathrm{th}$ root of unity in and extension F_{q^m}. Let $r = c + e$ be a received word with $c \in C$ and e an error polynomial of weight at most t. Let $J \subseteq \{0, 1, 2, \ldots, n - 1\}$ be the set of indices of the non-zero coefficients of e so that $e = \sum_{j \in J} e_j x^j$.

Decoding BCH Codes

Let C be a narrow sense ($b = 1$) BCH code length n and designed distance $d = 2t + 1$ over F_q, with generator polynomial g (we assume that q and n are relatively prime). Let α be a primitive $n\text{th}$ root of unity in and extension F_{q^m}. Let $r = c + e$ be a received word with $c \in C$ and e an error polynomial of weight at most t. Let $J \subseteq \{0, 1, 2, \dots, n-1\}$ be the set of indices of the non-zero coefficients of e so that $e = \sum_{j \in J} e_j x^j$.

The syndromes of r are defined as $h_i = r(\alpha^{i+1}) = e(\alpha^{i+1}) = \sum_{j \in J} e_j \alpha^{(i+1)j}$ for $0 \leq i \leq 2t-1$ and the **syndrome polynomial** is $h = \sum_{i=0}^{2t-1} h_i x^i$.

Error locator polynomial

The polynomial $\sigma = \prod_{j \in J}(1 - \alpha^j x)$ is called the **error locator polynomial** because the inverses of its roots give the locations $j \in J$ (i.e. if we know σ then we know the error locations.

Syndrome polynomial

The syndrome polynomial can be rewritten in the following form:

$$
\begin{aligned}
h &= \sum_{i=0}^{2t-1} \left(\sum_{j \in J} e_j \alpha^{(i+1)j} \right) x^i \\
&= \sum_{j \in J} \sum_{i=0}^{2t-1} e_j \alpha^{(i+1)j} x^i
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{j \in J} e_j \alpha^j \sum_{i=0}^{2t-1} (\alpha^j x)^i \\
&= \sum_{j \in J} \frac{e_j \alpha^j (1 - (\alpha^j x)^{2t})}{1 - \alpha^j x}.
\end{aligned}
$$

Multiplying this by σ we obtain

$$\sigma h = \sum_{j \in J} \left[e_j \alpha^j \prod_{\substack{k \in J \\ k \neq j}} (1 - \alpha^k x) \right] (1 - (\alpha^j x)^{2t})$$

and reduction modulo x^{2t} gives the congruence

$$\sigma h \equiv \sum_{j \in J} e_j \alpha^j \prod_{\substack{k \in J \\ k \neq j}} (1 - \alpha^k x) \bmod x^{2t}.$$

Error evaluator polynomial

The polynomial ω on the right hand side is known as the **error evaluator polynomial.** If we know σ and ω then we can calculate the values e_j of the errors.

Key equation

The congruence

$$\sigma h \equiv \omega \bmod x^{2t}$$

which is universally known as the **key equation** (after Berlekamp (1968)).

Error evaluator polynomial

The polynomial ω on the right hand side is known as the **error evaluator polynomial**. If we know σ and ω then we can calculate the values e_j of the errors.

Key equation

The congruence

$$\sigma h \equiv \omega \bmod x^{2t}$$

which is universally known as the **key equation** (after Berlekamp (1968)).

We seek a solution (σ, ω) with $deg(\sigma) \leq t$, $deg(\omega) \leq deg(\sigma)$ and σ, ω relatively prime.

Solution module

We define $M = \{(a, b) \mid ah \equiv b \bmod x^{2t}\}$ and call it the **solution module.**

Solution module

We define $M = \{(a, b) \mid ah \equiv b \bmod x^{2t}\}$ and call it the **solution module.**

Lemma

The set $\mathcal{B} = \{(1, h), (0, x^{2t})\}$ is a basis of M.
Proof. Obviously, $\mathcal{B} \subseteq M$. Now if $(a, b) \in M$ then $ah - b$ is a multiple of x^{2t} so
$(a, b) = a(1, h) - (0, ah - b) = a(1, h) + f(0, x^{2t})$.

A specific term order $<$ in $A = \mathbb{F}_q[x]^2$

$(1, 0) < (0, 1) < (x, 0) < (0, x) < \cdots$ is a term order in A so for example $(3x^2 - 2x + 1, 4x^3 + x - 5) = 4(0, x^3) + 3(x^2, 0) + (0, x) - 2(x, 0) - 5(0, 1) + (1, 0)$.

A specific term order $<$ in $A = \mathbb{F}_q[x]^2$

$(1,0) < (0,1) < (x,0) < (0,x) < \cdots$ is a term order in A so for example $(3x^2 - 2x + 1, 4x^3 + x - 5) =$
$4(0,x^3) + 3(x^2,0) + (0,x) - 2(x,0) - 5(0,1) + (1,0)$.

GB of a submodule $N \subseteq A$

Two possibilities for a GB of N.

$$N = \langle (a,b) \rangle$$

for some (a,b) where (a,b) is the minimal element of N.

$$N = \langle (a_1,b_1),(a_2,b_2) \rangle$$

where $lt(a_1,b_1) = (x^{p_1},0)$ with p_1 minimal, $lt(a_2,b_2) = (0,x^{p_2})$ with p_2 minimal, and either (a_1,b_1) or (a_2,b_2) is the minimal element of N.

Minimal element in solution module M

Theorem

If a solution (a, b) exists in M with $deg(a) \leq t, deg(b) \leq deg(a)$ and a, b relatively prime then (a, b) is the minimal element of M.

Minimal element in solution module M

Theorem

If a solution (a, b) exists in M with $deg(a) \leq t$, $deg(b) \leq deg(a)$ and a, b relatively prime then (a, b) is the minimal element of M.

Algorithm (PF)

Input: h, t
Output: $(a, b) \in M$ with $deg(a) \leq 2t$, $deg(b) \leq deg(a)$ and a, b relatively prime, if such an element exists
Initialize: $(a_1, b_1) := (1, h)$; $(a_2, b_2) := (0, x^{2t})$
WHILE $deg(a_1) \leq deg(b_1)$ DO [i.e. while $lt(a_1, b_1)$ on right]
  $(u, v) := (a_2, b_2)$ mod (a_1, b_1) [division algorithm]
  $(a_2, b_2) := (a_1, b_1)$
  $(a_1, b_1) := (u, v)$
$(a, b) := (a_1, b_1)$

Example: linear recurring sequence

Solution by approximations

For $k = 0, 1, \ldots 2t$ define $M_k = \{(a, b) \in A \mid ah \equiv b \bmod x^k\}$.

Solution by approximations

For $k = 0, 1, \ldots 2t$ define $M_k = \{(a, b) \in A \mid ah \equiv b \bmod x^k\}$.

Theorem (PF)

Let $\mathcal{B} = \{(a_1, b_1), (a_2, b_2)\}$ be a GB of M_k with (a_1, b_1) minimal and let

$$a_1 h - b_1 \equiv \alpha_1 x^k \bmod x^{k+1}$$

$$a_2 h - b_2 \equiv \alpha_2 x^k \bmod x^{k+1}.$$

Define $\mathcal{B}' = \{(a'_1, b'_1), (a'_2, b'_2)\}$ as follows.

If $\alpha_1 = 0$ then

$$(a'_1, b'_1) = (a_1, b_1), (a'_2, b'_2) = (xa_2, xb_2).$$

If $\alpha_1 \neq 0$ then

$$(a'_1, b'_1) = (xa_1, xb_1), (a'_2, b'_2) = (a_1, b_1) - \frac{\alpha_2}{\alpha_1}(a_2, b_2).$$

Then $\mathcal{B}'$ is a GB of M_{k+1}.

Algorithm (PF)

This theorem gives an obvious algorithm (which can be improved by suppressing the computation of the b_i).
The algorithm has the same complexity as Berlekamp-Massey.

Algorithm can be extended to list decoding algebraic geometry codes.

References

- See Fitzpatrick (*IEEE Trans IT 1995*) for the first paper to use this technique (simplified proof in notes)
- See chapter on coding theory in *Using Algebraic Geometry* by Cox, Little and O'Shea for a discussion in relation to RS codes.
- See Byrne and Fitzpatrick (*JSC 2000, IEEE Trans IT 2001*) for extension to codes over rings.
- See O'Keeffe and Fitzpatrick in *AAECC* journal (2006 or 2007) for extension to list decoding of algebraic geometry codes.