A Gröbner basis approach to list decoding of Reed-Solomon and Algebraic Geometry Codes.

Henry O'Keeffe and Patrick Fitzpatrick
Boole Centre for Research in Informatics,
National University of Ireland,Cork,
Ireland.
{*h.okeeffe,p.fitzpatrick*}*@ucc.ie*

"Gröbner Bases in Cryptography, Coding Theory, and Algebraic Combinatorics"

RICAM – Linz – Tuesday 2nd May 2006.

Overview

- List Decoding
- Sudan's algorithm
  - Reed-Solomon codes
  - 1–Point Algebraic Geometry codes
  - The algorithm and variations
  - module formulation for the interpolation step
  - soft decision
- Gröbner Basis module solution
  - Gröbner Bases for modules
  - general module algorithm/term orders
  - common decoding algorithm

Term Orders

Let F be any field and $A = F[x_1, \ldots, x_s]$ be the polynomial ring in s indeterminates over F. The *terms* of A are power products $x_1^{n_1}, \ldots, x_s^{n_s}$.

A^L is a free A-module and has a *standard basis* $\{\mathbf{e}_1, \ldots, \mathbf{e}_L\}$.

The terms of A^L are of the form

$$W\mathbf{e}_j, j \in [L], W \text{ a term of } A.$$

We define a *term order* $<$ on the terms of A^L as a total order with the following properties

$$\mathbf{X} < W\mathbf{X}, W \neq 1 \text{ a term of } A, \mathbf{X} \text{ any term of } A^L.$$

and

$$W\mathbf{X} < W\mathbf{Y}, W \text{ any term of } A, \mathbf{X}, \mathbf{Y} \text{ any terms of } A^L \text{ with } \mathbf{X} < \mathbf{Y}.$$

List decoding

For a block linear block code . . .

- Complete (*nearest neighbour*) decoding is an NP–complete problem
- If we assume a bound on the number of errors (*bounded decoding*)
  - does not exceed half the minimum distance
    - unique codeword produced
    - efficient algorithms exist for many codes (e.g. B-M, BMS)
  - otherwise
    - Elias (1957) and Wozencraft (1958)
    - codeword not always unique – hence a *list*
    - until recently, no efficient algorithms

Sudan's algorithm

Sudan (1997) presented a polynomial-time algorithm for the list decoding of (low rate) Reed-Solomon codes. It has two steps:-
1) find a polynomial by interpolation
2) factorise that polynomial to yield the list of valid codewords .

Its applicability was extended to (low rate) 1-point AG codes by Shokrollahi and Wasserman(1999).

Guruswami and Sudan (1999) enhanced the algorithm to address RS and AG codes of all rates.

Pellikaan and Wu (2004) shows that Reed-Müller codes of certain orders can be described by 1-point AG codes.

Reed Solomon codes

Let F_q be the finite field with q elements and $F_q[x]$ the ring of polynomials in one indeterminate over F_q. A Reed-Solomon code of dimension k and length $n = q - 1$ can viewed as the evaluation of polynomials in $F_q[x]$ with degree less than k at the n non-zero elements of F_q. We can define the Reed-Solomon code as the subspace

$$\mathcal{C}_q(n, k) = \{(f(\delta_1), \ldots, f(\delta_n)) | f \in F_q[x], \partial f < k\}.$$

1-point AG codes

Let χ be an absolutely irreducible curve of genus g over F_q. Denote the $n+1$ F_q-rational points on χ by $P_1, \ldots, P_n, P_\infty$ and the field of rational functions on χ by $F(\chi)$.
Define

- R_∞ the ring of elements of $F(\chi)$ with poles only at P_∞
- $\mathcal{L}(\ell P_\infty)$ the subset of R_∞ whose pole order at P_∞ is at most ℓ.

A 1-point AG code can be defined as the vector space (over F_q)

$$\mathcal{C}_\chi(\ell, P_\infty) = \{(f(P_1), \ldots, f(P_n)) | f \in \mathcal{L}(\ell P_\infty)\}.$$

(with dimension $\ell - g + 1$)

The received word and the interpolation step

Suppose $(c_1, \ldots, c_n)$ is transmitted and $(y_1, \ldots, y_n)$ is received.

- For a RS (*resp. AG*) code, each element of a codeword c_j takes value of a polynomial (*resp. rational function*) at a corresponding field element δ_j (*resp. rational point* P_j).
- The first step in Sudan–type list decoding involves finding a non-zero polynomial Q which pass through the points (y_j, δ_j) (*resp.* (y_j, P_j)) "m" times. The choice of *multiplicity* m and related contraints on the degree of Q guarantees that the polynomials (*resp. rational functions*) which generate the required codewords are to be found among the factors of Q.

Multiplicity – Reed Solomon code case

- $Q \in F_q[x, y]$ such that coefficients of $Q(x + \delta_j, y + y_j)$ are zero for terms of total degree less than m.

Multiplicity – Agebraic Geometry code case

- $\mathcal{K} = \bigcup_{r=0}^{\infty} \mathcal{L}(rP_\infty)$ is a field and $\mathcal{L}(\ell P_\infty)$ is a vector space of dimension $\ell - g + 1$ over F_q
  - At the point P_∞ there is a basis of functions $\phi_i \in \mathcal{K}$ with increasing pole order at P_∞.
  - At each rational point P_j there is a (vector space) basis of functions $\psi_{i,j} \in \mathcal{K}$ with increasing zero order at P_j.
- $Q \in \mathcal{K}[y]$, expanded around a basis with respect to P_∞, such that coefficients of $Q(y + y_j)$ expanded with respect to P_j are zero for terms $y^{j_1}\psi_{j_2,j}$ where $j_1 + (j_2 - 1) < m$.

(By inserting the zero function in the g "gaps", these bases can be extended to ones with $\ell + 1$ elements. As we shall see, the soft decision problem uses the latter and the hard decision the former)

“Degree” constraints

- RS
  - For $Q \in F_q[x, y]$ the (a, b)–degree(Q) is the maximum value of $ai + bj$ among all terms $x^i y^j$ with non-zero coefficients of Q.
  - A limit K on the $(1, k)$–degree(Q) combined with the multiplicity requirements ensure the existence of the interpolating polynomial.
- AG
  - $\alpha = k + g - 1$, $s = \lfloor \frac{\ell - g}{\alpha} \rfloor$ and $L = \ell - g + 1$
  - $Q[y]$ is required to have the form

$$Q[y] = \sum_{i=0}^{s} \sum_{j=1}^{L-\alpha i} q_{ij} \phi_j y^i$$

Module formulation

With these constraints, the solutions to the interpolations can be viewed as elements of free modules $F_q[x]^L$.

- RS
  - A limit on the $(1,k)$–degree(Q) means that the maximum value of the exponent of y is less than L for some $L \in N_0$.
  - On that subset of $F_q[x,y]$, define $\mu : F_q[x,y] \longrightarrow F_q[x]^L$

$$\mu(x^i y^j) = x^i \mathbf{e}_{j+1}$$

    and extend by linearity.
  - Define $H : F_q[x]^L \longrightarrow F_q[x]^L$

$$H(\mathbf{b}) = \mu(\mu^{-1}(\mathbf{b})(x+\delta_j, y+y_j))$$

- H is F_q linear and

$$H(x\mathbf{b}) = (x + \delta_j)H(\mathbf{b}).$$

- The transformed problem then seeks elements $\mathbf{b} \in F_q[x]^L$ where, for each interpolation "point",
  * all the terms $x^i\mathbf{e}_j$ of $\mathbf{b}$ satisfy $ik + (j-1) < K$
  * coefficients of $H(\mathbf{b})$ are zero for terms $x^i\mathbf{e}_j$ with $i + (j-1) < m$.

- AG
  - By associating ϕ_i with $\mathbf{e}_i$, we can view Q as an element Q_M of $F_q[y]^L$.
  - Similarly, by associating $\psi_{i,j}$ with $\mathbf{e}_i$, $Q(y + y_j)$ expanded at point P_j can be viewed as an element $Q_M^{(j,y_j)}$ of $F_q[y]^L$.
  - Define $H : F_q[y]^L \rightarrow F_q[y]^L$ as the function that maps Q_M to $Q_M^{(j,y_j)}$.

- H is F_q linear and

$$H(y\mathbf{b}) = (y + y_j)H(\mathbf{b}).$$

- The solutions sought are elements $\mathbf{b} \in F_q[y]^L$ where, for each interpolation "point",
  * all the terms $y^i\mathbf{e}_j$ of $\mathbf{b}$ satisfy $\alpha i + (j-1) < L$
  * coefficients of $H(\mathbf{b})$ are zero for terms $y^i\mathbf{e}_j$ with $i + (j-1) < m$.

- For single indeterminate z, $\langle\{z^i\mathbf{e}_j | i + (j-1) = m\}\rangle$ is a submodule of $F_q[z]^L$.

Soft–decision

Instead of a "hard" received word $(y_1, \ldots, y_n)$, the channel (or inner code) may present *reliability* information.

Kötter and Vardy (2000)
- a soft-decision list decoding algorithm
- Reed Solomon and 1-point AG codes
- modelled on Sudan's algorithm

Reliability to multiplicities

- An RS (*resp. AG*) code of length n defined over $F_q = \{\alpha_1, \ldots, \alpha_q\}$.
- Reliability information leads to a $q \times n$ *reliability matrix* Π of posterior probabilities

$$\pi_{ij} = Pr(\alpha_i \text{ sent}|y_j \text{ received}), i \in [q], j \in [n].$$

- (qn) multiplicities m_{ij} are derived from Π via a greedy algorithm.
- require Q to"pass through" the points (δ_j, α_i) (*resp.* (P_j, α_i)) m_{ij} times.
- This is a more general form but essentially the same problem as the hard information case.

A choice of term order for these problems

A term order $<_{c,w}$ of the module $F[z]^L$ can be defined by using a weight-vector $w = (w_1, \ldots, w_L) \in \mathbb{N}^L$ and $c \geq 1 \in \mathbb{N}$ as follows:

$$z^n \mathbf{e}_j <_w z^m \mathbf{e}_i$$

when $cn + w_j < cm + w_i$ or $cn + w_j = cm + w_i$ and $i < j$.

Ignoring the degree constraints, the solution set to these problems forms a submodule of $F_q[z]^L$. Since the existence of a solution satisfying the degree constraints is guaranteed, *a fortiori* a minimal element (contained in a Gröbner basis with respect an instance of this term order) will also be a solution.

Gröbner bases

The *leading term* of a module element $\mathbf{f}$ is the greatest of its terms with respect to the term order and will be denoted by $lt(\mathbf{f})$.

For M a submodule of A^L, let $< lt(M) >$ be the submodule generated by the leading terms of the elements of M

A set $G = \{\mathbf{g}_1, \ldots, \mathbf{g}_t\}$ is a Gröbner basis for M if $< lt(M) >=< lt(\mathbf{g}_1), \ldots, lt(\mathbf{g}_t) >$.

G has the following properties
- it is a generating set for M
- it contains an element which is minimal with respect to $<$.

A *strictly ordered* Gröbner basis is one which is ordered by the leading terms of its elements and those leading terms are strictly increasing.

The general problem

Again, let $A = F[x_1, \ldots, x_s]$. We seek solutions $\mathbf{b} \in A^L$ which satisfy a sequence of p congruences

$$H^{(k)}(\mathbf{b}) \equiv 0 \bmod M^{(k)}, k = 1, \ldots, p$$

where $M^{(k)}$ are A–modules.

Each $H^{(k)}$ is an F–linear function such that for each $i, 1 \leq i \leq s$ there exists $\gamma_i^{(k)} \in F$ satisfying

$$H^{(k)}(x_i\mathbf{b}) = (x_i + \gamma_i^{(k)})H^{(k)}(\mathbf{b})$$

for all $\mathbf{b} = (b_1, \ldots, b_L) \in A^L$. The solution set is a submodule of A^L.

The Module Sequence

Our general algorithm is applicable providing that for each $M^{(k)}$ we have a (descending) chain of modules

$$M_0^{(k)}, \ldots, M_\ell^{(k)}, \ldots, M_{N_k}^{(k)} = M^{(k)}$$

with $F-$homomorphisms θ_ℓ so that for each ℓ

$$M_\ell^{(k)} \supseteq M_{\ell+1}^{(k)}$$

$$\theta_\ell^{(k)} : M_\ell^{(k)} \to F, ker(\theta_\ell^{(k)}) = M_{\ell+1}^{(k)}. \tag{1}$$

As a consequence, there are constants $\beta_i^{(\ell,k)}$ where

$$(x_i - \beta_i^{(\ell,k)})M_\ell^{(k)} \subseteq M_{\ell+1}^{(k)}, 1 \leq i \leq s. \tag{2}$$

The Incremental Step

Theorem 1 *Let M be an A–module and let $M_\ell \supseteq M_{\ell+1}$ be submodules of M satisfying (1) for suitable θ_ℓ, β_i. Let $H : A^L \to M$ be an F–linear function such that for each $s, 1 \leq i \leq s$ there exists $\gamma_i \in F$ satisfying*

$$H(x_i \mathbf{b}) = (x_i + \gamma_i) H(\mathbf{b})$$

for all $\mathbf{b} = (b_1, \ldots, b_L) \in A^L$.
Let $S \subseteq A^L$ be a submodule satisfying

$$H(\mathbf{b}) \equiv 0 \bmod M_\ell \textit{ for all } \mathbf{b} \in S$$

and let $S' \subseteq S$ be the set of elements satisfying

$$H(\mathbf{b}) \equiv 0 \bmod M_{\ell+1}.$$

Then S' is a submodule of A^L.

If $\mathcal{W}$ is a strictly ordered Gröbner basis of S relative to a term order $<$ then a Gröbner basis $\mathcal{W}'$ of S' relative to $<$ can be constructed as follows

Define $\Delta_j := \theta_\ell(H(\mathcal{W}[j]))$ *for* $1 \leq j \leq |\mathcal{W}|$.
$\mathcal{W}' = \textit{incremental–step}(\mathcal{W}, [x_i], [\Delta_j], [\beta_i], [\gamma_i])$
Proc *incremental–step()*
If $\Delta_j = 0$ *for all* j *then*
$\mathcal{W}' = \mathcal{W}$
otherwise
$j^* :=$ *least* j *for which* $\Delta_j \neq 0$
$\mathcal{W}_1 := \{\mathcal{W}_j : j < j^*\}$
$\mathcal{W}_2 := \{(x_i - (\beta_i + \gamma_i))\mathcal{W}[j^*] : 1 \leq i \leq s\}$
$\mathcal{W}_3 := \{\mathcal{W}[j] - (\Delta_j/\Delta_{j^*})\mathcal{W}[j^*] : j > j^*\}$
$\mathcal{W}' := \mathcal{W}_1 \bigcup \mathcal{W}_2 \bigcup \mathcal{W}_3$
End

The Iterative Algorithm

- By ordering the output of the incremental step, the produced Gröbner basis can be used as the input to the next step.
- Any module $M_\ell^{(k)}$ can be chosen for the next step providing, of course, that the congruence $H^{(k)}(\mathbf{b}) \equiv 0 \bmod M_{\ell-1}^{(k)}$ has been processed by an earlier step.

- Let $T^{(i)} = T_{(j_1,\ldots,j_p)}$ be the submodule of A^L which satisfies

$$H^{(k)}(\mathbf{b}) \equiv 0 \bmod M_{j_k}^{(k)}, k = 1, \ldots, p$$

  and let $T^{(0)} = T_{(0,\ldots,0)}$ be an initial module for which a Gröbner basis is known. $T = T_{(N_1,\ldots,N_p)}$ is the submodule for which a Gröbner basis is sought.

- If $j_k \leq j'_k$ for all $k \in \{1, \ldots, p\}$ then $T_{(j_1,\ldots,j_p)} \supseteq T_{(j'_1,\ldots,j'_p)}$. In this way we can define a descending chain of modules $T^{(0)} \supseteq \cdots \supseteq T^{(j)} \supseteq \cdots \supseteq T$.

- Suppose that we have a strictly ordered Gröbner basis for $T^{(i)} = T_{(j_1,\ldots,j_p)}$. Then, providing $j'_k = j_k + 1$ for exactly one $k \in \{1, \ldots, p\}$, and $j'_k = j_k$ otherwise, the incremental step provides a Gröbner basis for $T^{(i+1)} = T_{(j'_1,\ldots,j'_p)}$. The resulting Gröbner basis can then converted into a strictly ordered Gröbner basis (by a function *ord*, say).

Input

functions $H^{(k)}$

constants $\gamma_i^{(k)}, 1 \leq k \leq p, 1 \leq i \leq s$

modules $M_\ell^{(k)}$ *and homomorphisms* $\theta_\ell^{(k)}$,

$1 \leq k \leq p, 0 \leq \ell \leq N_k$

constants $\beta_i^{(k,\ell)}$,

$1 \leq k \leq p, 1 \leq i \leq s, 0 \leq \ell \leq N_k$

$<$ *a term order on* A^L

$\mathcal{W}_0$ *a strictly ordered Gröbner basis of* $T^{(0)}$

Output

$\mathcal{W}$ *a strictly ordered Gröbner basis of the submodule* T

The function *nextmod* selects the next module in the decending chain i.e. sets up the input for the incremental step to find those elements of *module* which additionally satisfy

$$H^{(k)}(\mathbf{b}) \equiv 0 \bmod M_{\ell+1}^{(k)}.$$

Main Routine

$\mathcal{W} := \mathcal{W}_0$

For module *from* $T^{(0)}$ *to* T

$(k, \theta_\ell) =$*nextmod*(module)

$\Delta_j := \theta_\ell(H^{(k)}(\mathcal{W}[j]))$ *for* $j \in [|\mathcal{W}|]$

$\mathcal{W}' =$*incremental-step*$(\mathcal{W}, [x_i], [\Delta_j], \beta_i^{(k,\ell)}], \gamma_i^{(k)})$

$\mathcal{W} := \mathrm{ord}(\mathcal{W}')$

Initialisation

In these applications $M_0^{(k)} = A^L$. The standard basis of A^L, ordered with respect the chosen term order, will be the initial basis for the solutions to

$$H^{(k)}(\mathbf{b}) \equiv 0 \bmod M_0^{(k)}, k = 1, \ldots, p$$

Particular cases

- The interpolations have been transformed into congruences involving a single indeterminate.
- In the case of a single indeterminate, $F[z]$, and beginning with the standard basis, the number of elements $(=L)$ is unchanged at each step and *ord* is a simple function which merely inserts $\mathcal{W}_2$ into the correct location.
- If $\mathcal{W}_2$ exceeds the degree constraints, it can be dropped by the *ord* function and the size of the Gröbner basis could be reduced by 1.

The Incremental Step – single indeterminate case

When $A = F[z] \ldots$

Define $\Delta_j := \theta_\ell(H(\mathcal{W}[j]))$ for $1 \leq j \leq |\mathcal{W}|$.
$\mathcal{W}'$ =incremental–step$(\mathcal{W}, z, [\Delta_j], \beta, \gamma)$
Proc incremental–step()
If $\Delta_j = 0$ for all j then
$\mathcal{W}' = \mathcal{W}$
otherwise
$j^* :=$ least j for which $\Delta_j \neq 0$
$\mathcal{W}_1 := \{\mathcal{W}_j : j < j^*\}$
$\mathcal{W}_2 := \{(z - (\beta + \gamma))\mathcal{W}[j^*]\}$
$\mathcal{W}_3 := \{\mathcal{W}[j] - (\Delta_j / \Delta_{j^*})\mathcal{W}[j^*] : j > j^*\}$
$\mathcal{W}' := \mathcal{W}_1 \bigcup \mathcal{W}_2 \bigcup \mathcal{W}_3$
End

- While some of these interpolations could be solved directly by our algorithm, the transformed view results in more efficient algorithms
  - a homogeneous system of (linear) equations
  - the single indeterminate form has quadratic (vs. cubic) complexity
- If the set $\{z^i \mathbf{e}_j | i + (j-1) < m\}$ is ordered with respect to any term order, a sequence of modules begining with $F[z]^L$ and ending with $\langle \{z^i \mathbf{e}_j | i + (j-1) = m\} \rangle$ can be created such that

$$M_\ell = F z^i \mathbf{e}_j + M_{\ell+1}.$$

- From these we can define the functions θ_ℓ

$$\theta_\ell(\alpha z^i \mathbf{e}_j + \mathbf{a}) = \alpha, \mathbf{a} \in M_{\ell+1}.$$

- The constants β_i are all zero.

The Common Algorithm

Input

M the $q \times n$ multiplicity matrix

L the module dimension

Functions $H^{(j,\gamma_i)}, \gamma_i \in F_q, j \in [n]$.

c a weight for a term order $<_{c,(0,1,\ldots,L-1)}$

Output

The first element of $\mathcal{W}$, an ordered Gröbner basis

Main Routine

$\mathcal{W}$:=ord(the standard basis of $F_q[z]^L$).

For j from 1 to n

For i from 1 to q

If $m_{ij} \neq 0$

For j_2 from 1 to $min(L, m_{ij})$

For j_1 from 0 to $m_{ij} - j_2$

$\Delta_k :=$ coeff$(z^{j_1}\mathbf{e}_{j_2}, H^{(j,\gamma_i)}(\mathcal{W}[k]))$

for $k \in [L]$

$\mathcal{W}'$ =incremental–step$(\mathcal{W}, z, [\Delta_k], 0, \gamma_i)$

$\mathcal{W} :=$ ord$(\mathcal{W}')$

Specialisations of the common algorithm

- RS
  - The weight for the term order $c = k$.
- AG
  - The weight for the term order $c = k + g - 1$.
- Hard decision
  - $m_{ij} = m$ when $\gamma_j = y_j$
  - $m_{ij} = 0$ otherwise.

Questions ?