Gradient-like decoding of binary linear codes.

M. Borges-Quintana $^{(\star)}$

Joined work with: M. A. Borges-Trenard $^{(\star)}$ Edgar Martínez-Moro $^{(\star\star)}$

$(\star)$ Dpto. Matemática
Universidad de Oriente, Cuba

$(\star\star)$ Dpto. Matemática Aplicada
Universidad de Valladolid, Spain

July 5, 2008, S^{3}CM, Soria, España.

Part I, *Gröbner bases associated with binary linear codes*

Part II, *Gradient-like decoding of binary linear codes. Examples*

Part I

Gröbner bases associated with binary linear codes

Outline

Outline

Gröbner Bases and linear codes in the literature

To find a Gröbner basis of the error locator ideal.
A systematic method for encoding and decoding m-dimensional cyclic codes.
The key equation and Berlekam-Massey Algorithm (BCH codes).

Gröbner bases and the integer programing problem related with the soft-decision maximum likelihood decoding of binary linear block codes.

The FGLM algorithm has been connected already to coding theory:
Related with the error locator ideal and a change of ordering.
Related with the Berlekam-Massey Algorithm.

Gröbner Bases and linear codes in the literature

To find a Gröbner basis of the error locator ideal.

A systematic method for encoding and decoding m-dimensional cyclic codes.

The key equation and Berlekam-Massey Algorithm (BCH codes).

Gröbner bases and the integer programing problem related with the soft-decision maximum likelihood decoding of binary linear block codes.

The FGLM algorithm has been connected already to coding theory:

Related with the error locator ideal and a change of ordering.

Related with the Berlekam-Massey Algorithm.

Gröbner Bases and linear codes in the literature

To find a Gröbner basis of the error locator ideal.
A systematic method for encoding and decoding m-dimensional cyclic codes.
The key equation and Berlekam-Massey Algorithm (BCH codes).

Gröbner bases and the integer programing problem related with the soft-decision maximum likelihood decoding of binary linear block codes.

The FGLM algorithm has been connected already to coding theory:
Related with the error locator ideal and a change of ordering.
Related with the Berlekam-Massey Algorithm.

Gröbner Bases and linear codes in the literature

Also there are works devoted to extensions of the previous ideas to Algebraic Geometric Codes and codes over rings.

Gröbner bases and coding theory, an incomplete list of authors:
S. Sakata, T. Mora, M. Sala, E. Orsini, P. Fitzpatrick, J.C. Faugere, J. Fitzgerald, R.F. Lax, D. Ikegami, R. Pellikaan, S. Bulygin, ...

Our approach

(Borges-Quintana, Borges-Trenard, Martínez-Moro): a Gröbner basis is associated with the structure of the linear code.

Gröbner Bases and linear codes in the literature

Also there are works devoted to extensions of the previous ideas to Algebraic Geometric Codes and codes over rings.

Gröbner bases and coding theory, an incomplete list of authors:
S. Sakata, T. Mora, M. Sala, E. Orsini, P. Fitzpatrick, J.C. Faugere, J. Fitzgerald, R.F. Lax, D. Ikegami, R. Pellikaan, S. Bulygin, ...

Our approach

(Borges-Quintana, Borges-Trenard, Martínez-Moro): a Gröbner basis is associated with the structure of the linear code.

Gröbner Bases and linear codes in the literature

Also there are works devoted to extensions of the previous ideas to Algebraic Geometric Codes and codes over rings.

Gröbner bases and coding theory, an incomplete list of authors:
S. Sakata, T. Mora, M. Sala, E. Orsini, P. Fitzpatrick, J.C. Faugere, J. Fitzgerald, R.F. Lax, D. Ikegami, R. Pellikaan, S. Bulygin, ...

Our approach

(Borges-Quintana, Borges-Trenard, Martínez-Moro): a Gröbner basis is associated with the structure of the linear code.

Overview of the method

Let $\mathcal{A}$ be a finitely generated algebra (we want to solve a problem in $\mathcal{A}$).

- To find the appropiate morphism

$$\xi : K[X] \to \mathcal{A}.$$

$\mathcal{I}$ will be the ideal such that

$\mathcal{A} \cong K[X]/\mathcal{I} \cong Span_K(N)$ (N: the set of canonical forms).

If $\mathcal{A}$ is a monoid (or group) algebra ($A = K[M]$)

$$\mathcal{I} = \langle \{x^w - x^v \mid \xi(x^w) = \xi(x^v),\ x^w, x^v \in [X]\} \rangle.$$

Overview of the method

Let $\mathcal{A}$ be a finitely generated algebra (we want to solve a problem in $\mathcal{A}$).

- To find the appropiate morphism

$$\xi : K[X] \to \mathcal{A}.$$

$\mathcal{I}$ will be the ideal such that

$$\mathcal{A} \cong \mathbf{K[X]}/\mathcal{I} \cong \mathbf{Span_K(N)}\ \textbf{(N: the set of canonical forms).}$$

If $\mathcal{A}$ is a monoid (or group) algebra ($\mathbf{A} = \mathbf{K[M]}$)

$$\mathcal{I} = \langle \{ \mathbf{x^w} - \mathbf{x^v} \mid \xi(\mathbf{x^w}) = \xi(\mathbf{x^v}),\ \mathbf{x^w}, \mathbf{x^v} \in [\mathbf{X}] \} \rangle.$$

Overview of the method

Let $\mathcal{A}$ be a finitely generated algebra (we want to solve a problem in $\mathcal{A}$).

- To find the appropiate morphism

$$\xi : K[X] \to \mathcal{A}.$$

$\mathcal{I}$ will be the ideal such that

$\mathcal{A} \cong K[X]/\mathcal{I} \cong Span_K(N)$ (N: the set of canonical forms).

If $\mathcal{A}$ is a monoid (or group) algebra $(\mathbf{A} = \mathbf{K[M]})$

$$\mathcal{I} = \langle \{\mathbf{x^w} - \mathbf{x^v} \mid \xi(\mathbf{x^w}) = \xi(\mathbf{x^v}),\ \mathbf{x^w}, \mathbf{x^v} \in [\mathbf{X}]\} \rangle.$$

Overview of the method

$\xi : K[X] \rightarrow \mathcal{A}$,

$\mathcal{A} \cong K[X]/\mathcal{I} \cong Span_K(N)$ (N: the set of canonical forms)

- To find or construct an appropiate ordering $\prec$ on $[X]$.
- Define a reduction process (s.t. it allows to solve the problem):
  - ⋆ finite numbers of reductions,
  - ⋆ unique canonical forms.

Overview of the method

$\xi : K[X] \rightarrow \mathcal{A}$,

$\mathcal{A} \cong K[X]/\mathcal{I} \cong Span_K(N)$ (N: the set of canonical forms)

- To find or construct an appropiate ordering $\prec$ on $[X]$.
- Define a reduction process (s.t. it allows to solve the problem):
  - ⋆ finite numbers of reductions,
  - ⋆ unique canonical forms.

Overview of the method

$\xi : K[X] \rightarrow \mathcal{A}$,

$\mathcal{A} \cong K[X]/\mathcal{I} \cong Span_K(N)$ (N: the set of canonical forms)

- To find or construct an appropiate ordering $\prec$ on $[X]$.
- Define a reduction process (s.t. it allows to solve the problem):
  - ⋆ **finite numbers of reductions,**
  - ⋆ **unique canonical forms.**

Overview of the method: the instance of linear codes

Having:

$$\xi : K[X] \rightarrow \mathcal{A} = K[M],$$

$$K[M] \cong K[X]/\mathcal{I} \cong Span_K(N) \text{ (N: the set of canonical forms)}$$

The instance of linear codes:

1. $M = \mathbb{F}_2^{n-k}$ (the monoids of the syndromes).
2. ξ: gives the syndrome of each $x^w \in [X]$.
3. $N \Leftrightarrow$ the coset leaders.
4. $\mathcal{I}$: we call it the ideal associated with the code.

Then, we compute a Gröbner basis of $\mathcal{I}$ for a convenient ordering $\prec$ such that:

$$Can(x^w, I, \prec) \Leftrightarrow \text{the coset leader with syndrome } \xi(x^w).$$

Overview of the method: the instance of linear codes

Having:

$$\xi : K[X] \rightarrow \mathcal{A} = K[M],$$

$$K[M] \cong K[X]/\mathcal{I} \cong Span_K(N) \text{ } (N\text{: the set of canonical forms})$$

The instance of linear codes:

1. $M = \mathbb{F}_2^{n-k}$ (the monoids of the syndromes).
2. ξ: gives the syndrome of each $x^w \in [X]$.
3. $N \Leftrightarrow$ the coset leaders.
4. $\mathcal{I}$: we call it the ideal associated with the code.

Then, we compute a Gröbner basis of $\mathcal{I}$ for a convenient ordering $\prec$ such that:

$$Can(x^w, I, \prec) \Leftrightarrow \text{the coset leader with syndrome } \xi(x^w).$$

Overview of the method: the instance of linear codes

Having:

$$\xi : K[X] \rightarrow \mathcal{A} = K[M],$$

$$K[M] \cong K[X]/\mathcal{I} \cong Span_K(N) \text{ } (N\text{: the set of canonical forms})$$

The instance of linear codes:

1. $M = \mathbb{F}_2^{n-k}$ (the monoids of the syndromes).
2. ξ: gives the syndrome of each $x^w \in [X]$.
3. $N \Leftrightarrow$ the coset leaders.
4. $\mathcal{I}$: we call it the ideal associated with the code.

Then, we compute a Gröbner basis of $\mathcal{I}$ for a convenient ordering $\prec$ such that:

$$Can(x^w, I, \prec) \Leftrightarrow \text{the coset leader with syndrome } \xi(x^w).$$

Buchberger's Algorithm

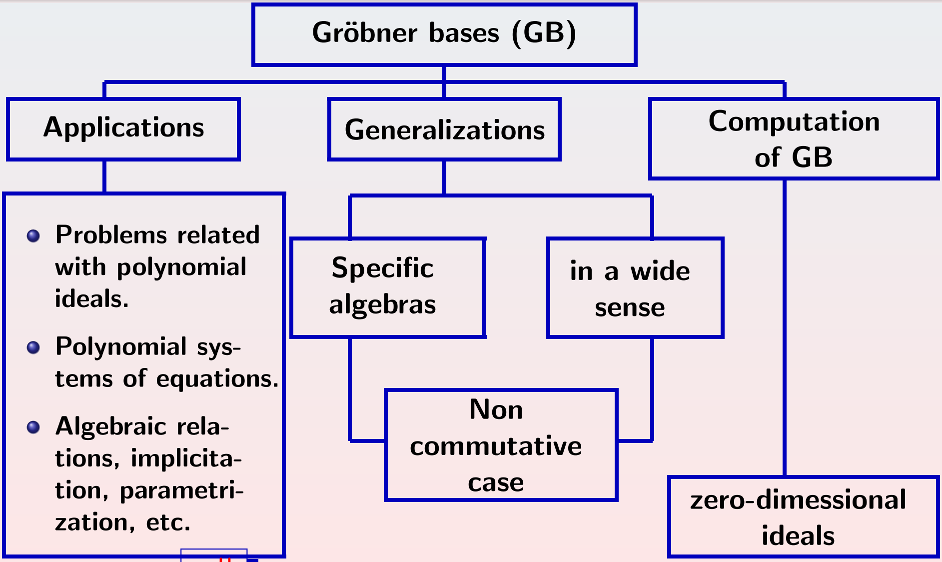
Input: A set of polynomials F s.t.

$$I = Ideal(F),$$

$\prec$ an **admissible** ordering on $[X]$.

Output: Gröbner basis of I w.r.t. $\prec$.

Ejemplo


Gröbner bases (GB)
Applications
Generalizations
Computation of GB
Problems related with polynomial ideals.
Polynomial systems of equations.
Algebraic relations, implicitation, parametrization, etc.
Specific algebras
in a wide sense
Non commutative case
zero-dimessional ideals

Outline

Computation of GB

Specialization to specific algebras | **Using other inputs**

zero-dimenssional ideals

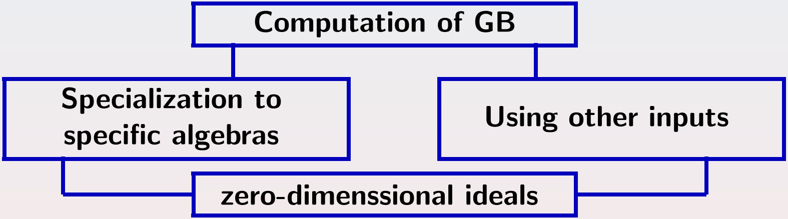
zero-dimenssional: $\dim K\langle X\rangle/I < \infty$

Linear Algebra in $K\langle X\rangle/I$:

$$\forall c_i \in K,\ s_i \in \langle X\rangle \qquad \sum_{i=1}^{r} c_i s_i \in I \setminus \{0\} \Leftrightarrow$$

$\{s_1, \ldots, s_r\}$, is linear dependent module I.

More . . .



zero-dimenssional: $\dim K\langle X\rangle / I < \infty$

Linear Algebra in $K\langle X\rangle / I$:

$$\forall c_i \in K,\ s_i \in \langle X\rangle \qquad \sum_{i=1}^{r} c_i s_i \in I \setminus \{0\} \Leftrightarrow$$

$\{s_1, \ldots, s_r\}$, is linear dependent module I.

More . . .

Outline

1. Gröbner Bases
2. The zero-dimenssional ideal case
3. Binary Codes
   - Introduction
   - A monoid representation of $\mathbb{F}_2^n$
4. Some comments about specialized algorithms
   - Considering the code as a monoid
   - Starting from a generator matrix

A primer on binary codes

Let $\mathbb{F}_2$ be the finite field of two elements. A binary code $\mathcal{C}$ of dimension k and length n is the image of a linear (injective) mapping:

$$L : \mathbb{F}_2^k \longrightarrow \mathbb{F}_2^n \quad k \leq n.$$

There exists an $n \times (n-k)$ matrix H called parity check matrix such that for each word (element) we have: $\mathbf{c} \cdot H = \mathbf{0} \Leftrightarrow c \in \mathcal{C}$.

A primer on binary codes

Let $\mathbb{F}_2$ be the finite field of two elements. A binary code $\mathcal{C}$ of dimension k and length n is the image of a linear (injective) mapping:

$$L : \mathbb{F}_2^k \longrightarrow \mathbb{F}_2^n \quad k \leq n.$$

There exists an $n \times (n-k)$ matrix H called parity check matrix such that for each word (element) we have: $\mathbf{c} \cdot H = \mathbf{0} \Leftrightarrow c \in \mathcal{C}$.

A primer on binary codes (Cont.)

The weight of a word is its Hamming distance to the word $\mathbf{0}$, i.e. the number of non-zero coordinates of the word. The minimun distance d of the code $\mathcal{C}$ is the minimum weight among all the non-zero codewords.

The error-correcting capacity of a code is $t = \lfloor \frac{d-1}{2} \rfloor$. If we let $B(\mathcal{C}, t) := \{y \in \mathbb{F}_2^n \mid \exists c \in \mathcal{C} \text{ s.t. } d(y, c) \leq t\}$, it is well known that the equation $e \cdot H = y \cdot H$ has a unique solution e with $\mathrm{weight}(e) \leq t$ if $y \in B(\mathcal{C}, t)$. Then $y - e \in \mathcal{C}$ (syndrome decoding).

A primer on binary codes (Cont.)

The weight of a word is its Hamming distance to the word $\mathbf{0}$, i.e. the number of non-zero coordinates of the word. The minimun distance d of the code $\mathcal{C}$ is the minimum weight among all the non-zero codewords.

The error-correcting capacity of a code is $t = \lfloor \frac{d-1}{2} \rfloor$. If we let $B(\mathcal{C}, t) := \{y \in \mathbb{F}_2^n \mid \exists c \in \mathcal{C} \text{ s.t. } d(y, c) \leq t\}$, it is well known that the equation $e \cdot H = y \cdot H$ has a unique solution e with $\mathrm{weight}(e) \leq t$ if $y \in B(\mathcal{C}, t)$. Then $y - e \in \mathcal{C}$ (syndrome decoding).

A monoid representation

Let us consider the free commutative monoid $[X]$ generated by the n variables $X := \{x_1, \ldots, x_n\}$.

We have the following map from X to $\mathbb{F}_2^n$:

$$\psi : X \to \mathbb{F}_2^n, \text{ where } x_i \mapsto e_i \text{ (the } i\text{-th coordinate vector).}$$

The map ψ can be extended in a natural way to a morphism from $[X]$ onto $\mathbb{F}_2^n$, where $\psi(\prod_{i=1}^n x_i^{\beta_i}) = (\beta_1 \bmod 2, \ldots, \beta_n \bmod 2)$.

A monoid representation

Let us consider the free commutative monoid $[X]$ generated by the n variables $X := \{x_1, \ldots, x_n\}$.

We have the following map from X to $\mathbb{F}_2^n$:

$$\psi : X \longrightarrow \mathbb{F}_2^n, \text{ where } x_i \mapsto e_i \text{ (the } i\text{-th coordinate vector).}$$

The map ψ can be extended in a natural way to a morphism from $[X]$ onto $\mathbb{F}_2^n$, where $\psi(\prod_{i=1}^n x_i^{\beta_i}) = (\beta_1 \bmod 2, \ldots, \beta_n \bmod 2)$.

A monoid representation

Let us consider the free commutative monoid $[X]$ generated by the n variables $X := \{x_1, \ldots, x_n\}$.

We have the following map from X to $\mathbb{F}_2^n$:

$$\psi : X \to \mathbb{F}_2^n, \text{ where } x_i \mapsto e_i \text{ (the } i\text{-th coordinate vector).}$$

The map ψ can be extended in a natural way to a morphism from $[X]$ onto $\mathbb{F}_2^n$, where $\psi(\prod_{i=1}^n x_i^{\beta_i}) = (\beta_1 \bmod 2, \ldots, \beta_n \bmod 2)$.

The equivalence relation $R_{\mathcal{C}}$ in $\mathbb{F}_2^n$

A binary code $\mathcal{C}$ defines an equivalence relation $R_{\mathcal{C}}$ in $\mathbb{F}_2^n$:

$$(x, y) \in R_{\mathcal{C}} \Leftrightarrow x - y \in \mathcal{C}. \tag{1}$$

Let $\xi(x^u) := \psi(x^u)H$ (note that $x^u \in [X]$). The above congruence can be translated to $[X]$ by the linear morphims ξ as

$$x^u \cong_{\mathcal{C}} x^w \Leftrightarrow (\psi(x^u), \psi(x^w)) \in R_{\mathcal{C}} \Leftrightarrow \xi(x^u) = \xi(x^w). \tag{2}$$

The equivalence relation $R_{\mathcal{C}}$ in $\mathbb{F}_2^n$

A binary code $\mathcal{C}$ defines an equivalence relation $R_{\mathcal{C}}$ in $\mathbb{F}_2^n$:

$$(x, y) \in R_{\mathcal{C}} \Leftrightarrow x - y \in \mathcal{C}. \quad (1)$$

Let $\xi(x^u) := \psi(x^u)H$ (note that $x^u \in [X]$). The above congruence can be translated to $[X]$ by the linear morphims ξ as

$$x^u \cong_{\mathcal{C}} x^w \Leftrightarrow (\psi(x^u), \psi(x^w)) \in R_{\mathcal{C}} \Leftrightarrow \xi(x^u) = \xi(x^w). \quad (2)$$

The binomial ideal associated with the code

Let $I(\mathcal{C})$ be the ideal associated with the relation $R_{\mathcal{C}}$ on $[X]$, that is:

$$\mathbf{I}(\mathcal{C}) := \langle \{ \mathbf{x}^{\mathbf{w}} - \mathbf{x}^{\mathbf{v}} \mid (\psi(\mathbf{x}^{\mathbf{u}}), \psi(\mathbf{x}^{\mathbf{w}})) \in \mathbf{R}_{\mathcal{C}} \} \rangle.$$

Outline

Computing the reduced Gröbner basis

Let G_{T} be the reduced Gröbner basis of the ideal $I(\mathcal{C})$ with respect to $<$ (a total degree compatible ordering).

There are diferent algorithmic ways of computing G_{T} for this setting.

FGLM algorithm for monoid algebras

FGLM algorithm for finite dimension monoid algebras: An algorithm based on the generation of representative elements of the quotient algebra w.r.t. a given admisible ordering (in this case $\mathbb{F}_2^{n-k} \cong K[X]/I(\mathcal{C})$) based on linear algebra techniques. ([4], specialized to linear codes in [1].)

Outputs: G_{T}. It can also give a set N of representative elements corresponding to a set of coset leaders for the code and a matrix structure that allows to multiply the representative elements (to sum the cosets leaders and obtaining the corresponding coset leader).

Skip details . . .

FGLM algorithm for monoid algebras

FGLM algorithm for finite dimension monoid algebras: An algorithm based on the generation of representative elements of the quotient algebra w.r.t. a given admisible ordering (in this case $\mathbb{F}_2^{n-k} \cong K[X]/I(\mathcal{C})$) based on linear algebra techniques. ([4], specialized to linear codes in [1].)

Outputs: $\mathbf{G_T}$. It can also give a set N of representative elements corresponding to a set of coset leaders for the code and a matrix structure that allows to multiply the representative elements (to sum the cosets leaders and obtaining the corresponding coset leader).

Skip details . . .

Monoid rings

Let M be a finite commutative monoid generated by $g_1, \ldots, g_n$;

$\xi : [X] \longrightarrow M$, the canonical morphism that sends x_i to g_i;

$\sigma \subset [X] \times [X]$, a presentation of M defined by ξ

$$\sigma = \{(x^w, x^v) \mid \xi(x^w) = \xi(x^v)\}.$$

Monoid rings

$\xi : [X] \to M$, the canonical morphism that sends x_i to g_i;

$\sigma \subset [X] \times [X]$, a presentation of M defined by ξ

$$\sigma = \{(x^w, x^v) \mid \xi(x^w) = \xi(x^v)\}.$$

Then, it is known that the monoid ring $K[M]$ is isomorphic to $K[X]/I(\sigma)$, where $I(\sigma)$ is the ideal generated by $P(\sigma) = \{x^w - x^v \mid (x^w, x^v) \in \sigma\}$

$$I(\sigma) = \langle P(\sigma)\rangle = \langle\{x^w - x^v \mid (x^w, x^v) \in \sigma\}\rangle.$$

Moreover, any Gröbner basis G of $I(\sigma)$ is also formed by binomials of the above form. **In addition, it can be proved that $\{(\mathbf{x^w}, \mathbf{x^v}) \mid \mathbf{x^w} - \mathbf{x^v} \in \mathbf{G}\}$ is another presentation of M.**

Monoid rings

$\xi : [X] \longrightarrow M$, the canonical morphism that sends x_i to g_i;

$\sigma \subset [X] \times [X]$, a presentation of M defined by ξ

$$\sigma = \{(x^w, x^v) \mid \xi(x^w) = \xi(x^v)\}.$$

Then, it is known that the monoid ring $K[M]$ is isomorphic to $K[X]/I(\sigma)$, where $I(\sigma)$ is the ideal generated by $P(\sigma) = \{x^w - x^v \mid (x^w, x^v) \in \sigma\}$

$$I(\sigma) = \langle P(\sigma) \rangle = \langle \{x^w - x^v \mid (x^w, x^v) \in \sigma\} \rangle.$$

Moreover, any Gröbner basis G of $I(\sigma)$ is also formed by binomials of the above form. **In addition, it can be proved that $\{(\mathbf{x^w}, \mathbf{x^v}) \mid \mathbf{x^w} - \mathbf{x^v} \in \mathbf{G}\}$ is another presentation of $\mathbf{M}$.**

Monoid rings

$\xi : [X] \rightarrow M$, the canonical morphism that sends x_i to g_i;

$\sigma \subset [X] \times [X]$, a presentation of M defined by ξ

$$\sigma = \{(x^w, x^v) \mid \xi(x^w) = \xi(x^v)\}.$$

Then, it is known that the monoid ring $K[M]$ is isomorphic to $K[X]/I(\sigma)$, where $I(\sigma)$ is the ideal generated by $P(\sigma) = \{x^w - x^v \mid (x^w, x^v) \in \sigma\}$

$$I(\sigma) = \langle P(\sigma) \rangle = \langle \{x^w - x^v \mid (x^w, x^v) \in \sigma\} \rangle.$$

Moreover, any Gröbner basis G of $I(\sigma)$ is also formed by binomials of the above form. **In addition, it can be proved that $\{(\mathbf{x^w}, \mathbf{x^v}) \mid \mathbf{x^w} - \mathbf{x^v} \in \mathbf{G}\}$ is another presentation of M.**

Monoid rings

Note that M is finite if and only if $I = I(\sigma)$ is zero-dimensional.

Specifying the monoid M

The monoid M is set to be $\mathbb{F}_2^{n-k}$ (where the syndromes belong to).

Doing $g_i := \xi(x_i)$, note that

$$\mathbf{M} = \mathbb{F}_2^{\mathbf{n-k}} = \langle \mathbf{g_1}, \ldots, \mathbf{g_n} \rangle.$$

Moreover, $\sigma := \mathrm{R}_{\mathcal{C}}$, hence $\mathrm{I}(\sigma) = \mathrm{I}(\mathcal{C})$.

Specifying the monoid M

The monoid M is set to be $\mathbb{F}_2^{n-k}$ (where the syndromes belong to).

Doing $g_i := \xi(x_i)$, note that

$$\mathbf{M} = \mathbb{F}_2^{\mathbf{n-k}} = \langle \mathbf{g_1}, \ldots, \mathbf{g_n} \rangle.$$

Moreover, $\sigma := \mathrm{R}_{\mathcal{C}}$, hence $\mathrm{I}(\sigma) = \mathrm{I}(\mathcal{C})$.

Starting from a generator matrix

Starting from a generator matrix.

Starting from a generator matrix

Obtaining the ideal I($\mathcal{C}$): Let be $\{w_1, \ldots, w_k\}$ be the row vectors of a generator matrix for a code (more generally any matrix whose rows span the code $\mathcal{C}$), i.e., a basis (spanning set) of the code as subspace of $\mathbb{F}_2^n$ (see [3]). Let

$$I = \langle \{x^{w_1} - 1, \ldots, x^{w_k} - 1\} \cup \{x_i^2 - 1 \mid i = 1, \ldots, n\} \rangle \qquad (3)$$

be the ideal generated by the set of binomials $\{x^{w_1} - 1, \ldots, x^{w_k} - 1\} \cup \{x_i^2 - 1 \mid i = 1, \ldots, n\} \subset K[X]$. **Since $\{\mathrm{w}_1, \ldots, \mathrm{w}_k\}$ generates $\mathcal{C}$ it is clear that $\mathrm{I} = \mathrm{I}(\mathcal{C})$.**

Starting from a generator matrix

Obtaining the ideal $\mathbf{I}(\mathcal{C})$: Let be $\{w_1, \ldots, w_k\}$ be the row vectors of a generator matrix for a code (more generally any matrix whose rows span the code $\mathcal{C}$), i.e., a basis (spanning set) of the code as subspace of $\mathbb{F}_2^n$ (see [3]). Let

$$I = \langle \{x^{w_1} - 1, \ldots, x^{w_k} - 1\} \cup \{x_i^2 - 1 \mid i = 1, \ldots, n\} \rangle \qquad (3)$$

be the ideal generated by the set of binomials $\{x^{w_1} - 1, \ldots, x^{w_k} - 1\} \cup \{x_i^2 - 1 \mid i = 1, \ldots, n\} \subset K[X]$. **Since $\{\mathbf{w_1}, \ldots, \mathbf{w_k}\}$ generates $\mathcal{C}$ it is clear that $\mathbf{I} = \mathbf{I}(\mathcal{C})$.**

Starting from a generator matrix

Let $F = \{x^{w_1} - 1, \ldots, x^{w_k} - 1\} \cup \{x_i^2 - 1 \mid i = 1, \ldots, n\} \subset K[X]$, $r = k + n$. There are two ways for computing G_T:

Starting from a generator matrix

Let $\mathbf{F} = \{x^{w_1} - 1, \ldots, x^{w_k} - 1\} \cup \{x_i^2 - 1 \mid i = 1, \ldots, n\} \subset K[X]$, $r = k + n$. There are two ways for computing G_T:

- G_T can be computed by **Buchberger's algorithm** starting with the initial set F.
  However, there are some computational advantages in this case. The coefficient field is $\mathbb{F}_2$ (therefore, there is no coefficient growth), and the maximal length of words in the computation is n (the binomials $x_i^2 - 1$ prevent the length being greater than n).

Starting from a generator matrix

Let $F = \{x^{w_1} - 1, \ldots, x^{w_k} - 1\} \cup \{x_i^2 - 1 \mid i = 1, \ldots, n\} \subset K[X]$, $r = k + n$. There are two ways for computing G_{T}:

- G_{T} can be computed by **Buchberger's algorithm** starting with the initial set F.
  However, there are some computational advantages in this case. The coefficient field is $\mathbb{F}_2$ (therefore, there is no coefficient growth), and the maximal length of words in the computation is n (the binomials $x_i^2 - 1$ prevent the length being greater than n). Thus the two principal disadvantages of Gröbner basis computations are not valid for this case. In addition, total degree compatible term orders are among the most efficient for the computation of Gröbner bases.

Starting from a generator matrix

Let $F = \{x^{w_1}-1,\ldots,x^{w_k}-1\} \cup \{x_i^2-1 \mid i=1,\ldots,n\} \subset K[X]$, $r = k+n$. There are two ways for computing G_T:

- **Using the FGLM basis convertion algorithm** to obtain a basis for the syzygy module M in $K[X]^{r+1}$ of the generator set $F' = \{-1, f_1, f_2, \ldots, f_r\}$. Each of the syzygies corresponds to a solution

$$f = \sum_{i=1}^{r} b_i f_i \qquad b_i \in K[X], i = 1, \ldots, r.$$

  and thus points to an element f in the ideal I generated by F.

Part II

Gradient-like decoding of binary linear codes. Examples

Outline

5. Gradient decoding of binary codes
   - Test sets
6. Gröbner test sets
7. Worked example
   - other structures (Matphi and border basis)
8. Remarks and Complexity

Minimal subsets in codes

Let $\mathbb{F}_2^n$ the n-dimensional coordinate space over the field $\mathbb{F}_2$ and $\mathcal{C} \subseteq \mathbb{F}_2^n$ a linear code. We define the **support of a codeword** $\mathbf{c} = (c_1, \ldots, c_n) \in \mathcal{C}$ as

$$\mathrm{supp}(\mathbf{c}) = \{i \in \{1, \ldots, n\} \mid c_i \neq 0\}\,. \tag{4}$$

If $\mathrm{supp}(\mathbf{c}') \subset \mathrm{supp}(\mathbf{c})$ (respectively $\subseteq$) we will write $\mathbf{c}' \prec \mathbf{c}$ (respectively $\preceq$).

Definition (Minimal codeword)

A nonzero vector $\mathbf{c} \in \mathcal{C}$ is said to be **minimal** if $0 \neq \mathbf{c}' \preceq \mathbf{c}$ and $\mathbf{c}' \in \mathcal{C}$ then it implies that there exists a nonzero constant $\alpha \in \mathbb{F}_2$ such that $\mathbf{c}' = \alpha\mathbf{c}$.

Minimal subsets in codes

Let $\mathbb{F}_2^n$ the n-dimensional coordinate space over the field $\mathbb{F}_2$ and $\mathcal{C} \subseteq \mathbb{F}_2^n$ a linear code. We define the **support of a codeword** $\mathbf{c} = (c_1, \ldots, c_n) \in \mathcal{C}$ as

$$\operatorname{supp}(\mathbf{c}) = \{i \in \{1, \ldots, n\} \mid c_i \neq 0\}. \tag{4}$$

If $\operatorname{supp}(\mathbf{c}') \subset \operatorname{supp}(\mathbf{c})$ (respectively $\subseteq$) we will write $\mathbf{c}' \prec \mathbf{c}$ (respectively $\preceq$).

Definition (Minimal codeword)

A nonzero vector $\mathbf{c} \in \mathcal{C}$ is said to be **minimal** if $0 \neq \mathbf{c}' \preceq \mathbf{c}$ and $\mathbf{c}' \in \mathcal{C}$ then it implies that there exists a nonzero constant $\alpha \in \mathbb{F}_2$ such that $\mathbf{c}' = \alpha \mathbf{c}$.

Test set gradient-like decoding

Definition

Any set $\mathcal{T} \subseteq \mathcal{C}$ of codewords such that for every vector $\mathbf{y} \in \mathbb{F}_2^n$ either $\mathbf{y}$ lies in $\mathcal{C}$ or there exists $\mathbf{z} \in \mathcal{T}$ such that

$$\mathrm{d}(\mathbf{y}+\mathbf{z},\mathbf{0}) < \mathrm{d}(\mathbf{y},\mathbf{0})$$

is called a **test set**.

Note that the set of minimal codewords is a test sets.

Test set gradient-like decoding

Definition

Any set $\mathcal{T} \subseteq \mathcal{C}$ of codewords such that for every vector $\mathbf{y} \in \mathbb{F}_2^n$ either $\mathbf{y}$ lies in $\mathcal{C}$ or there exists $\mathbf{z} \in \mathcal{T}$ such that

$$\mathrm{d}(\mathbf{y}+\mathbf{z},\mathbf{0}) < \mathrm{d}(\mathbf{y},\mathbf{0})$$

is called a **test set**.

Note that the set of minimal codewords is a test sets.

A **gradient-like** decoding algorithm is obtained using a test set $\mathcal{T}$ as follows.

Let $\mathbf{y}$ be the received vector

1. $\mathbf{c} \leftarrow \mathbf{0}$,
2. Find $\mathbf{z} \in \mathcal{T}$ such that $\mathrm{d}(\mathbf{y}+\mathbf{z}, \mathbf{0}) < \mathrm{d}(\mathbf{y}, \mathbf{0})$.

$$\mathbf{c} \leftarrow \mathbf{c}+\mathbf{z} \text{ and } \mathbf{y} \leftarrow \mathbf{y}+\mathbf{z}.$$

3. Repeat 2. until no such $\mathbf{z}$ is found in $\mathcal{T}$.
4. Return $\mathbf{c}$.

This decoding algorithm always converges to one of the closest codewords to the received vector.

Outline

Let $[X] = \{\mathbf{x}^{\mathbf{a}} \mid \mathbf{a} \in \mathbb{N}^n\}$ be the set of terms.

Let G be the reduced Gröbner basis of the ideal $I(\mathcal{C})$ with respect to the term ordering $<$ (a total degree compatible ordering) and let $g \in K[X]$, we denote by $\mathrm{Can}(g, G)$ the **canonical form** of g with respect to the Gröbner basis G.

Theorem (GB's Reduction means decoding)

Let $\mathcal{C}$ be a linear code. Let $x^w \in [X]$ and $x^v \in N$ its corresponding canonical form. If $\mathrm{weight}(\psi(x^v)) \leq t$ then $\psi(x^v)$ is the error vector corresponding to $\psi(x^w)$. Otherwise, if $\mathrm{weight}(\psi(x^v)) > t$, $\psi(x^w)$ contains more than t errors. (t is the error correcting capability)

Let $[X] = \{\mathbf{x}^{\mathbf{a}} \mid \mathbf{a} \in \mathbb{N}^n\}$ be the set of terms.

Let G be the reduced Gröbner basis of the ideal $I(\mathcal{C})$ with respect to the term ordering $<$ (a total degree compatible ordering) and let $g \in K[X]$, we denote by $\mathrm{Can}(g, G)$ the **canonical form** of g with respect to the Gröbner basis G.

Theorem (GB's Reduction means decoding)

Let $\mathcal{C}$ be a linear code. Let $x^w \in [X]$ and $x^v \in N$ its corresponding canonical form. If $\mathrm{weight}(\psi(x^v)) \leq t$ then $\psi(x^v)$ is the error vector corresponding to $\psi(x^w)$. Otherwise, if $\mathrm{weight}(\psi(x^v)) > t$, $\psi(x^w)$ contains more than t errors. (t is the error correcting capability)

Let $[X] = \{\mathbf{x}^{\mathbf{a}} \mid \mathbf{a} \in \mathbb{N}^n\}$ be the set of terms.

Let G be the reduced Gröbner basis of the ideal $I(\mathcal{C})$ with respect to the term ordering $<$ (a total degree compatible ordering) and let $g \in K[X]$, we denote by $\mathrm{Can}(g, G)$ the **canonical form** of g with respect to the Gröbner basis G.

Theorem (GB's Reduction means decoding)

Let $\mathcal{C}$ be a linear code. Let $x^w \in [X]$ and $x^v \in N$ its corresponding canonical form. If $\mathrm{weight}(\psi(x^v)) \leq t$ then $\psi(x^v)$ is the error vector corresponding to $\psi(x^w)$. Otherwise, if $\mathrm{weight}(\psi(x^v)) > t$, $\psi(x^w)$ contains more than t errors. (t is the error correcting capability)

If $g = \mathbf{x}^{\mathbf{w}} - \mathbf{x}^{\mathbf{v}} \in I(\mathcal{C})$ denote by $\mathbf{c}_g$ the codeword associated to the binomial g, that is,

$$\mathbf{c}_g = \psi(\mathbf{x}^{\mathbf{w}}) + \psi(\mathbf{x}^{\mathbf{v}})$$

Definition

Let G be a Gröbner basis with respect to the term ordering $<$ of the binomial ideal $I(\mathcal{C})$. The **Gröbner codewords set** w.r.t. G is

$$\mathcal{C}_G = \{\mathbf{c}_g \mid g \in G\} \setminus \{\mathbf{0}\}.$$

If $g = \mathbf{x}^{\mathbf{w}} - \mathbf{x}^{\mathbf{v}} \in I(\mathcal{C})$ denote by $\mathbf{c}_g$ the codeword associated to the binomial g, that is,

$$\mathbf{c}_g = \psi(\mathbf{x}^{\mathbf{w}}) + \psi(\mathbf{x}^{\mathbf{v}})$$

Definition

Let G be a Gröbner basis with respect to the term ordering $<$ of the binomial ideal $I(\mathcal{C})$. The **Gröbner codewords set** w.r.t. G is

$$\mathcal{C}_G = \{\mathbf{c}_g \mid g \in G\} \setminus \{\mathbf{0}\}.$$

Theorem

The elements of the set $\mathcal{C}_G$ of Gröbner codewords are minimal codewords of the code $\mathcal{C}$.

Not every minimal codeword is a Gröbner codeword! Not even for a border basis of $\mathcal{C}$.

Theorem

The elements of the set $\mathcal{C}_G$ of Gröbner codewords are minimal codewords of the code $\mathcal{C}$.

Not every minimal codeword is a Gröbner codeword! Not even for a border basis of $\mathcal{C}$.

The Gröbner decoding algorithm

The theorem allows us to perform a gradient-like decoding algorithm but according to $<$ instead of the weight of the vectors. Thus we say that the set of Gröbner codewords is a "test set".

Input: $\mathcal{C}_G$ and $\mathbf{y}$ a received vector.
Output: One of the closest codewords to $\mathbf{y}$.

1. $i := 0$; $\mathbf{v}_i = \mathbf{y}$; $\mathbf{c}_i = \mathbf{0}$.
2. Repeat
3. Find $\mathbf{w} \in \mathcal{C}_G$ such that $\mathbf{x}^{\mathbf{v}_i} > \mathbf{x}^{\mathbf{v}_{i+1}}$ and $\mathbf{v}_{i+1} = \mathbf{v}_i + \mathbf{w}$.
4. $\mathbf{c}_{i+1} = \mathbf{c}_i + \mathbf{w}$; $i = i + 1$
5. Until such a $\mathbf{w}$ does not exist.
6. Return$[\mathbf{c}_i]$.

The Gröbner decoding algorithm

The theorem allows us to perform a gradient-like decoding algorithm but according to $<$ instead of the weight of the vectors. Thus we say that the set of Gröbner codewords is a "test set".

Input: $\mathcal{C}_G$ and $\mathbf{y}$ a received vector.
Output: One of the closest codewords to $\mathbf{y}$.

1. $i := 0$; $\mathbf{v}_i = \mathbf{y}$; $\mathbf{c}_i = \mathbf{0}$.
2. Repeat
3. Find $\mathbf{w} \in \mathcal{C}_G$ such that $\mathbf{x}^{\mathbf{v}_i} > \mathbf{x}^{\mathbf{v}_{i+1}}$ and $\mathbf{v}_{i+1} = \mathbf{v}_i + \mathbf{w}$.
4. $\mathbf{c}_{i+1} = \mathbf{c}_i + \mathbf{w}$; $i = i + 1$
5. Until such a $\mathbf{w}$ does not exist.
6. Return$[\mathbf{c}_i]$.

Outline

Worked example

Consider the code $\mathcal{C}$ in $\mathbb{F}_2^6$ (a $[6,3,3]$ binary linear code) with generator matrix:

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

The set of codewords is

$$\begin{aligned} \mathcal{C} \;=\; & \{(0,0,0,0,0,0),(1,0,1,1,0,0), \\ & (1,1,0,0,1,0),(0,1,0,1,0,1), \\ & (0,0,1,0,1,1),(1,1,1,0,0,1), \\ & (0,1,1,1,1,0),(1,0,0,1,1,1)\} \end{aligned}$$

Worked example

Consider the code $\mathcal{C}$ in $\mathbb{F}_2^6$ (a $[6,3,3]$ binary linear code) with generator matrix:

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

The set of codewords is

$$\begin{aligned} \mathcal{C} &= \{(0,0,0,0,0,0),(1,0,1,1,0,0), \\ &\quad (1,1,0,0,1,0),(0,1,0,1,0,1), \\ &\quad (0,0,1,0,1,1),(1,1,1,0,0,1), \\ &\quad (0,1,1,1,1,0),(1,0,0,1,1,1)\} \end{aligned}$$

The reduced Gröbner basis of $I(\mathcal{C})$ with respect to the degree reverse Lexicographical ordering $<$ is

$$\begin{aligned} G = \{ & x_1^2 - 1, x_2^2 - 1, x_3^2 - 1, x_4^2 - 1, x_5^2 - 1, x_6^2 - 1, \\ & x_1x_2 - x_5, x_1x_3 - x_4, x_1x_4 - x_3, x_1x_5 - x_2, \\ & x_2x_3 - x_1x_6, x_2x_4 - x_6, x_2x_5 - x_1, x_2x_6 - x_4, \\ & x_3x_4 - x_1, x_3x_5 - x_6, x_3x_6 - x_5, \\ & x_4x_5 - x_1x_6, x_4x_6 - x_2, x_5x_6 - x_3 \}. \end{aligned}$$

Therefore the code is 1-correcting (i.e. $t = 1$) and the set of Gröbner codewords is

$$\mathcal{C}_G = \left\{ \begin{array}{l} (1,1,0,0,1,0), (1,0,1,1,0,0), (0,1,0,1,0,1), \\ (0,0,1,0,1,1), (1,1,1,0,0,1), (1,0,0,1,1,1) \end{array} \right\}$$

The reduced Gröbner basis of $I(\mathcal{C})$ with respect to the degree reverse Lexicographical ordering $<$ is

$$\begin{aligned} G = \{ & x_1^2-1, x_2^2-1, x_3^2-1, x_4^2-1, x_5^2-1, x_6^2-1, \\ & x_1x_2-x_5, x_1x_3-x_4, x_1x_4-x_3, x_1x_5-x_2, \\ & x_2x_3-x_1x_6, x_2x_4-x_6, x_2x_5-x_1, x_2x_6-x_4, \\ & x_3x_4-x_1, x_3x_5-x_6, x_3x_6-x_5, \\ & x_4x_5-x_1x_6, x_4x_6-x_2, x_5x_6-x_3\}. \end{aligned}$$

Therefore the code is 1-correcting (i.e. $t=1$) and the set of Gröbner codewords is

$$\mathcal{C}_G = \left\{ \begin{array}{l} (1,1,0,0,1,0),(1,0,1,1,0,0),(0,1,0,1,0,1), \\ (0,0,1,0,1,1),(1,1,1,0,0,1),(1,0,0,1,1,1) \end{array} \right\}$$

The reduced Gröbner basis of $I(\mathcal{C})$ with respect to the degree reverse Lexicographical ordering $<$ is

$$\begin{aligned} G = \{ & x_1^2-1, x_2^2-1, x_3^2-1, x_4^2-1, x_5^2-1, x_6^2-1, \\ & x_1x_2-x_5, x_1x_3-x_4, x_1x_4-x_3, x_1x_5-x_2, \\ & x_2x_3-x_1x_6, x_2x_4-x_6, x_2x_5-x_1, x_2x_6-x_4, \\ & x_3x_4-x_1, x_3x_5-x_6, x_3x_6-x_5, \\ & x_4x_5-x_1x_6, x_4x_6-x_2, x_5x_6-x_3 \}. \end{aligned}$$

Therefore the code is 1-correcting (i.e. $t=1$) and the set of Gröbner codewords is

$$\mathcal{C}_G = \left\{ \begin{array}{c} (1,1,0,0,1,0),(1,0,1,1,0,0),(0,1,0,1,0,1), \\ (0,0,1,0,1,1),(1,1,1,0,0,1),(1,0,0,1,1,1) \end{array} \right\}$$

The reduced Gröbner basis of $I(\mathcal{C})$ with respect to the degree reverse Lexicographical ordering $<$ is

$$\begin{aligned} G = \{ & x_1^2-1, x_2^2-1, x_3^2-1, x_4^2-1, x_5^2-1, x_6^2-1, \\ & x_1x_2-x_5, x_1x_3-x_4, x_1x_4-x_3, x_1x_5-x_2, \\ & x_2x_3-x_1x_6, x_2x_4-x_6, x_2x_5-x_1, x_2x_6-x_4, \\ & x_3x_4-x_1, x_3x_5-x_6, x_3x_6-x_5, \\ & x_4x_5-x_1x_6, x_4x_6-x_2, x_5x_6-x_3 \}. \end{aligned}$$

Therefore the code is 1-correcting (i.e. $t=1$) and the set of Gröbner codewords is

$$\mathcal{C}_G = \left\{ \begin{array}{l} (1,1,0,0,1,0),(1,0,1,1,0,0),(0,1,0,1,0,1), \\ (0,0,1,0,1,1),(1,1,1,0,0,1),(1,0,0,1,1,1) \end{array} \right\}$$

1. If we recive $\mathbf{y} = (1,1,0,1,1,0)$; then

$$\mathbf{c}_1 := (1,1,0,0,1,0) \text{ and } \mathbf{y}_1 = \mathbf{y} + \mathbf{c}_1 = (0,0,0,1,0,0).$$

Since $\mathrm{d}(\mathbf{y}_1, \mathbf{0}) = 1$, i.e. the codeword corresponding to $\mathbf{y}$ is $\mathbf{c}_1$.

2. Let $\mathbf{y} = (1,1,0,1,0,0)$; then

$$\mathbf{c}_1 := (0,1,0,1,0,1) \text{ and } \mathbf{y}_1 = \mathbf{y} + \mathbf{c}_1 = (1,0,0,0,0,1).$$

$\mathbf{y}_1$ can not be reduced following the algorithm; thus, $\mathrm{d}(\mathbf{y}_1, \mathbf{0}) > 1$; and in this case $\mathbf{y}$ contains more errors than the error-correcting capability of the code. However, note that $\mathbf{c}_1$ is the closest codeword to $\mathbf{y}$.

Skip . . .

1. If we recive $\mathbf{y} = (1,1,0,1,1,0)$; then

$$\mathbf{c}_1 := (1,1,0,0,1,0) \text{ and } \mathbf{y}_1 = \mathbf{y} + \mathbf{c}_1 = (0,0,0,1,0,0).$$

   Since $\mathrm{d}(\mathbf{y}_1, \mathbf{0}) = 1$, i.e. the codeword corresponding to $\mathbf{y}$ is $\mathbf{c}_1$.

2. Let $\mathbf{y} = (1,1,0,1,0,0)$; then

$$\mathbf{c}_1 := (0,1,0,1,0,1) \text{ and } \mathbf{y}_1 = \mathbf{y} + \mathbf{c}_1 = (1,0,0,0,0,1).$$

   $\mathbf{y}_1$ can not be reduced following the algorithm; thus, $\mathrm{d}(\mathbf{y}_1, \mathbf{0}) > 1$; and in this case $\mathbf{y}$ contains more errors than the error-correcting capability of the code. However, note that $\mathbf{c}_1$ is the closest codeword to $\mathbf{y}$.

Skip . . .

Example: Other outputs

$N = \{1, x_1, x_2, x_3, x_4, x_5, x_6, x_1x_6\}$;
Matphi (Practical representation):

$$
\begin{array}{ccccl}
 & & & & \qquad\qquad\quad \psi(w) \qquad\qquad\qquad {\color{red}\phi(w, x_i)} \\
 & & & & \qquad\qquad\quad \downarrow \qquad\qquad\qquad\quad {\color{red}\downarrow} \\
w & \longrightarrow & 1 & \longrightarrow & [\,[[0,0,0,0,0,0],\ 1,\ {\color{red}[2,3,4,5,6,7]}\,], \\
 & & x_1 & \longrightarrow & \ \ [[1,0,0,0,0,0],\ 1,\ {\color{red}[1,6,5,4,3,8]}\,], \\
 & & x_2 & \longrightarrow & \ \ [[0,1,0,0,0,0],\ 1,\ {\color{red}[6,1,8,7,2,5]}\,], \\
 & & x_3 & \longrightarrow & \ \ [[0,0,1,0,0,0],\ 1,\ {\color{red}[5,8,1,2,7,6]}\,], \\
 & & x_4 & \longrightarrow & \ \ [[0,0,0,1,0,0],\ 1,\ {\color{red}[4,7,2,1,8,3]}\,], \\
 & & x_5 & \longrightarrow & \ \ [[0,0,0,0,1,0],\ 1,\ {\color{red}[3,2,7,8,1,4]}\,], \\
 & & x_6 & \longrightarrow & \ \ [[0,0,0,0,0,1],\ 1,\ {\color{red}[8,5,6,3,4,1]}\,], \\
 & & x_2x_3 & \longrightarrow & \ \ [[0,1,1,0,0,0],\ 0,\ {\color{red}[7,4,3,6,5,2]}\,]\,]
\end{array}
$$

Example: Decoding

(i.) $y \in B(C,t)$: $y = (1,1,0,1,1,0)$; $w_y := x_1x_2x_4x_5$; $\phi(1,x_1) = x_1$; $\phi(x_1,x_2) = x_5$; $\phi(x_5,x_4) = x_2x_3$; $\phi(x_2x_3,x_5) = x_4$, this means $e = \psi(x_4) = (0,0,0,1,0,0)$, $weight(e) = 1$ then, the codeword corresponding to y is $c = y - e = (1,1,0,0,1,0)$.

(ii.) $y \notin B(C,t)$: $y = (0,1,0,0,1,1)$; $w_y := x_2x_5x_6$; $\phi(1,x2) = x_2$; $\phi(x_2,x_5) = x_1$; $\phi(x_1,x_6) = x_2x_3$; $e = (0,1,1,0,0,0)$; $weight(e) > 1$ then, we report an error in the transmission process.

Example: Decoding

(i.) $y \in B(C,t)$: $y = (1,1,0,1,1,0)$; $w_y := x_1x_2x_4x_5$; $\boldsymbol{\phi(1, x_1) = x_1}$; $\boldsymbol{\phi(x_1, x_2) = x_5}$; $\boldsymbol{\phi(x_5, x_4) = x_2x_3}$; $\boldsymbol{\phi(x_2x_3, x_5) = x_4}$, **this means $e = \phi(x_4) = (0,0,0,1,0,0)$, weight$(e) = 1$ then, the codeword corresponding to y is $c = y - e = (1,1,0,0,1,0)$.**

(ii.) $y \notin B(C,t)$: $y = (0,1,0,0,1,1)$; $w_y := x_2x_5x_6$; $\phi(1, x2) = x_2$; $\phi(x_2, x_5) = x_1$; $\phi(x_1, x_6) = x_2x_3$; $e = (0,1,1,0,0,0)$; $weight(e) > 1$ then, we report an error in the transmission process.

Example: Decoding

(i.) $y \in B(C,t)$: $y = (1,1,0,1,1,0)$; $w_y := x_1x_2x_4x_5$; $\phi(1,x_1) = x_1$; $\boldsymbol{\phi(x_1, x_2) = x_5}$; $\phi(x_5, x_4) = x_2x_3$; $\phi(x_2x_3, x_5) = x_4$, **this means** $e = \psi(x_4) = (0,0,0,1,0,0)$, **weight(e) = 1 then, the codeword corresponding to y is** $c = y - e = (1,1,0,0,1,0)$.

(ii.) $y \notin B(C,t)$: $y = (0,1,0,0,1,1)$; $w_y := x_2x_5x_6$; $\phi(1,x2) = x_2$; $\phi(x_2,x_5) = x_1$; $\phi(x_1,x_6) = x_2x_3$; $e = (0,1,1,0,0,0)$; $weight(e) > 1$ then, we report an error in the transmission process.

Example: Decoding

(i.) $y \in B(C,t)$: $y = (1,1,0,1,1,0)$; $w_y := x_1x_2x_4x_5$; $\phi(1,x_1) = x_1$; $\phi(x_1,x_2) = x_5$; $\boldsymbol{\phi(x_5,x_4) = x_2x_3}$; $\phi(x_2x_3, x_5) = x_4$, this means $e = \psi(x_4) = (0,0,0,1,0,0)$, weight$(e) = 1$ then, the codeword corresponding to y is $c = y - e = (1,1,0,0,1,0)$.

(ii.) $y \notin B(C,t)$: $y = (0,1,0,0,1,1)$; $w_y := x_2x_5x_6$; $\phi(1,x2) = x_2$; $\phi(x_2,x_5) = x_1$; $\phi(x_1,x_6) = x_2x_3$; $e = (0,1,1,0,0,0)$; $weight(e) > 1$ then, we report an error in the transmission process.

Example: Decoding

(i.) $y \in B(C,t)$: $y = (1,1,0,1,1,0)$; $w_y := x_1x_2x_4x_5$; $\phi(1,x_1) = x_1$; $\phi(x_1,x_2) = x_5$; $\phi(x_5,x_4) = x_2x_3$; $\boldsymbol{\phi(x_2x_3,x_5) = x_4}$, this means $e = \psi(x_4) = (0,0,0,1,0,0)$, weight$(e) = 1$ then, the codeword corresponding to y is $c = y - e = (1,1,0,0,1,0)$.

(ii.) $y \notin B(C,t)$: $y = (0,1,0,0,1,1)$; $w_y := x_2x_5x_6$; $\phi(1,x2) = x_2$; $\phi(x_2,x_5) = x_1$; $\phi(x_1,x_6) = x_2x_3$; $e = (0,1,1,0,0,0)$; $weight(e) > 1$ then, we report an error in the transmission process.

Example: Decoding

(i.) $y \in B(C,t)$: $y = (1,1,0,1,1,0)$; $w_y := x_1x_2x_4x_5$; $\phi(1,x_1) = x_1$; $\phi(x_1,x_2) = x_5$; $\phi(x_5,x_4) = x_2x_3$; $\phi(x_2x_3,x_5) = x_4$, **this means $\mathbf{e} = \psi(\mathbf{x_4}) = (0,0,0,1,0,0)$, weight(e) $= 1$ then,** the codeword corresponding to y is $c = y - e = (1,1,0,0,1,0)$.

(ii.) $y \notin B(C,t)$: $y = (0,1,0,0,1,1)$; $w_y := x_2x_5x_6$; $\phi(1,x2) = x_2$; $\phi(x_2,x_5) = x_1$; $\phi(x_1,x_6) = x_2x_3$; $e = (0,1,1,0,0,0)$; $weight(e) > 1$ then, we report an error in the transmission process.

Example: Decoding

(i.) $y \in B(C,t)$: $y = (1,1,0,1,1,0)$; $w_y := x_1x_2x_4x_5$; $\phi(1,x_1) = x_1$; $\phi(x_1,x_2) = x_5$; $\phi(x_5,x_4) = x_2x_3$; $\phi(x_2x_3,x_5) = x_4$, this means $e = \psi(x_4) = (0,0,0,1,0,0)$, $weight(e) = 1$ then, **the codeword corresponding to y is $\mathbf{c = y - e = (1,1,0,0,1,0)}$.**

(ii.) $y \notin B(C,t)$: $y = (0,1,0,0,1,1)$; $w_y := x_2x_5x_6$; $\phi(1,x2) = x_2$; $\phi(x_2,x_5) = x_1$; $\phi(x_1,x_6) = x_2x_3$; $e = (0,1,1,0,0,0)$; $weight(e) > 1$ then, we report an error in the transmission process.

Example: Decoding

(i.) $y \in B(C,t)$: $y = (1,1,0,1,1,0)$; $w_y := x_1x_2x_4x_5$; $\phi(1,x_1) = x_1$; $\phi(x_1,x_2) = x_5$; $\phi(x_5,x_4) = x_2x_3$; $\phi(x_2x_3,x_5) = x_4$, this means $e = \psi(x_4) = (0,0,0,1,0,\ 0)$, $weight(e) = 1$ then, the codeword corresponding to y is $c = y - e = (1,1,0,0,1,0)$.

(ii.) $y \notin B(C,t)$: $y = (0,1,0,0,1,1)$; $w_y := x_2x_5x_6$; $\phi(1,x2) = x_2$; $\phi(x_2,x_5) = x_1$; $\phi(x_1,x_6) = x_2x_3$; $e = (0,1,1,0,0,0)$; $weight(e) > 1$ then, we report an error in the transmission process.

Outline

Some computations with the binary Golay Code

We use the **GAP** package **GUAVA** to construct a generator matrix of the binary Golay code $[23, 12, 7]$ and **GBLA_LC**: a group of programs made in **GAP** to carry out our approach.

The code has 4096 codewords, 2048 syndromes.

1. Computing the reduced Gröbner basis (**Gr**): 17.42 min.
2. Computing the border basis (**BB**): 13.34 min.
3. $\mid \mathbf{Gr} \mid = \mathbf{8878}$.
4. $\mid \mathbf{BB} \mid = \mathbf{14697}$.
5. $\mathcal{C}_{\mathbf{Gr}} = \mathcal{C}_{\mathbf{BB}}$ and $\mid \mathcal{C}_{\mathbf{Gr}} \mid = \mathbf{253}$.

Some computations with the binary Golay Code

We use the **GAP** package **GUAVA** to construct a generator matrix of the binary Golay code $[23, 12, 7]$ and **GBLA_LC**: a group of programs made in **GAP** to carry out our approach.

The code has 4096 codewords, 2048 syndromes.

1. **Computing the reduced Gröbner basis (Gr): 17.42 min.**
2. Computing the border basis (**BB**): 13.34 min.
3. $\mid \mathbf{Gr} \mid = \mathbf{8878}$.
4. $\mid \mathbf{BB} \mid = \mathbf{14697}$.
5. $\mathcal{C}_{\mathbf{Gr}} = \mathcal{C}_{\mathbf{BB}}$ and $\mid \mathcal{C}_{\mathbf{Gr}} \mid = \mathbf{253}$.

Some computations with the binary Golay Code

We use the **GAP** package **GUAVA** to construct a generator matrix of the binary Golay code $[23, 12, 7]$ and **GBLA_LC**: a group of programs made in **GAP** to carry out our approach.

The code has 4096 codewords, 2048 syndromes.

1. Computing the reduced Gröbner basis (**Gr**): 17.42 min.
2. **Computing the border basis (BB): 13.34 min.**
3. $\mid \mathbf{Gr} \mid = \mathbf{8878}$.
4. $\mid \mathbf{BB} \mid = \mathbf{14697}$.
5. $\mathcal{C}_{\mathbf{Gr}} = \mathcal{C}_{\mathbf{BB}}$ and $\mid \mathcal{C}_{\mathbf{Gr}} \mid = \mathbf{253}$.

Some computations with the binary Golay Code

We use the **GAP** package **GUAVA** to construct a generator matrix of the binary Golay code $[23, 12, 7]$ and **GBLA_LC**: a group of programs made in **GAP** to carry out our approach.

The code has 4096 codewords, 2048 syndromes.

1. Computing the reduced Gröbner basis (**Gr**): 17.42 min.
2. Computing the border basis (**BB**): 13.34 min.
3. **$\mid$ Gr $\mid = 8878$.**
4. $\mid$ **BB** $\mid = $ **14697**.
5. $\mathcal{C}_{\mathbf{Gr}} = \mathcal{C}_{\mathbf{BB}}$ and $\mid \mathcal{C}_{\mathbf{Gr}} \mid = $ **253**.

Some computations with the binary Golay Code

We use the **GAP** package **GUAVA** to construct a generator matrix of the binary Golay code $[23, 12, 7]$ and **GBLA_LC**: a group of programs made in **GAP** to carry out our approach.

The code has 4096 codewords, 2048 syndromes.

1. Computing the reduced Gröbner basis (**Gr**): 17.42 min.
2. Computing the border basis (**BB**): 13.34 min.
3. $\mid \mathbf{Gr} \mid = \mathbf{8878}$.
4. $\mid \mathbf{BB} \mid = \mathbf{14697.}$
5. $\mathcal{C}_{\mathbf{Gr}} = \mathcal{C}_{\mathbf{BB}}$ and $\mid \mathcal{C}_{\mathbf{Gr}} \mid = \mathbf{253}$.

Some computations with the binary Golay Code

We use the **GAP** package **GUAVA** to construct a generator matrix of the binary Golay code $[23, 12, 7]$ and **GBLA_LC**: a group of programs made in **GAP** to carry out our approach.

The code has 4096 codewords, 2048 syndromes.

1. Computing the reduced Gröbner basis (**Gr**): 17.42 min.
2. Computing the border basis (**BB**): 13.34 min.
3. $\mid \mathbf{Gr} \mid = \mathbf{8878}$.
4. $\mid \mathbf{BB} \mid = \mathbf{14697}$.
5. $\mathcal{C}_{\mathbf{Gr}} = \mathcal{C}_{\mathbf{BB}}$ **and** $\mid \mathcal{C}_{\mathbf{Gr}} \mid = \mathbf{253}$.

Some computations with the binary Golay Code

Computing the reduced Gröbner basis in some system of Symbolic Computation:

- **Mathematica (4.0)**: was interrupted after 4 hours.
- **Maple (9.0)**: was interrupted after 4 hours.
- **Singular (3-0-0)**: it succeeded in 2 hours.

- The decoding procedure is a complete decoding procedure, that is, it always finds the codeword that is the closest to the received vector.
- Furthermore, with the same procedure it is easy to know whether the result is reliable or not, if $\mathrm{d}(\mathbf{v}_i, \mathbf{0}) \leq t$ then $\mathbf{c}_i$ is the codeword corresponding to $\mathbf{y}$, where t is the error-correcting capability of $\mathcal{C}$. As a byproduct of the computation of $\mathcal{C}_G$ it is possible to obtain t.
- Generalizations to linear codes is possible by using the border basis, the extension of the ordering "the error vector ordering" (not anymore a degree compatible ordering) is not admissible.

- The decoding procedure is a complete decoding procedure, that is, it always finds the codeword that is the closest to the received vector.
- Furthermore, with the same procedure it is easy to know whether the result is reliable or not, if $\mathrm{d}(\mathbf{v}_i, \mathbf{0}) \leq t$ then $\mathbf{c}_i$ is the codeword corresponding to $\mathbf{y}$, where t is the error-correcting capability of $\mathcal{C}$. As a byproduct of the computation of $\mathcal{C}_G$ it is possible to obtain t.
- Generalizations to linear codes is possible by using the border basis, the extension of the ordering "the error vector ordering" (not anymore a degree compatible ordering) is not admissible.

- The decoding procedure is a complete decoding procedure, that is, it always finds the codeword that is the closest to the received vector.
- Furthermore, with the same procedure it is easy to know whether the result is reliable or not, if $\mathrm{d}(\mathbf{v}_i, \mathbf{0}) \leq t$ then $\mathbf{c}_i$ is the codeword corresponding to $\mathbf{y}$, where t is the error-correcting capability of $\mathcal{C}$. As a byproduct of the computation of $\mathcal{C}_G$ it is possible to obtain t.
- Generalizations to linear codes is possible by using the border basis, the extension of the ordering "the error vector ordering" (not anymore a degree compatible ordering) is not admissible.

Complexity

Preprocesing Computing the reduced Gröbner basis or the border basis performed $\mathcal{O}(n^2 2^{n-k})$ operations.

Decoding The decoding complexity depends on the size of $\mathcal{C}_G$ (or $\mathcal{C}_{\mathcal{B}}$) and the number of reductions. The number of reductions for $\mathcal{C}_{\mathcal{B}}$ is less than n.

Computing t The error correction capability of an arbitrary linear code (not neccesary binary) can be computed in at most $m \cdot n \cdot S(t+1)$ itererations of the Algorithm showed in B.,B. & M. where

$$S(l) = \sum_{i=0}^{l} \binom{n}{i} (q-1)^i.$$

Complexity

Preprocesing Computing the reduced Gröbner basis or the border basis performed $\mathcal{O}(n^2 2^{n-k})$ operations.

Decoding The decoding complexity depends on the size of $\mathcal{C}_G$ (or $\mathcal{C}_\mathcal{B}$) and the number of reductions. The number of reductions for $\mathcal{C}_\mathcal{B}$ is less than n.

Computing t The error correction capability of an arbitrary linear code (not neccesary binary) can be computed in at most $m \cdot n \cdot S(t+1)$ itererations of the Algorithm showed in B.,B. & M. where

$$S(l) = \sum_{i=0}^{l} \binom{n}{i} (q-1)^i.$$

Complexity

Preprocesing Computing the reduced Gröbner basis or the border basis performed $\mathcal{O}(n^2 2^{n-k})$ operations.

Decoding The decoding complexity depends on the size of $\mathcal{C}_G$ (or $\mathcal{C}_{\mathcal{B}}$) and the number of reductions. The number of reductions for $\mathcal{C}_{\mathcal{B}}$ is less than n.

Computing $\mathbf{t}$ The error correction capability of an arbitrary linear code (not neccesary binary) can be computed in at most $m \cdot n \cdot S(t+1)$ itererations of the Algorithm showed in B.,B. & M. where

$$S(l) = \sum_{i=0}^{l} \binom{n}{i} (q-1)^i.$$

Part III

References

References:

- J. Abbott, M. Kreuzer, L. Robiano. Computing Zero-Dimensional Schemes. *J. Symb. Comp.* 39(1), p. 31-49, 2005.
- W.W. Adams, Ph. Loustaunau. *An introduction to Gröbner bases*. Graduate Studies in Mathematics, 3. American Mathematical Society, Providence, RI, 1994.
- M.E. Alonso, M.G. Marinari, T. Mora. The Big Mother of all Dualities: Möller Algorithm. *Comm. Algebra* 31(2), 783-818, 2003.
- M. A. Borges-Trenard, M. Borges-Quintana, and T. Mora. Computing Gröbner bases by FGLM techniques in a non-commutative setting. *J. Symb. Comp.* 30(4), p. 429–449, 2000.

References:

M. Borges-Quintana, M. Borges-Trenard and E. Martínez-Moro. On a Gröbner bases structure associated to linear codes. J. Discrete Math. Sci. Cryptogr 10(2), p. 151-191, 2007.

M. Borges-Quintana, M. A. Borges-Trenard and E. Martínez-Moro. GBLA-LC: Gröbner basis by linear algebra and codes. International Congress of Mathematicians 2006 (Madrid), Mathematical Software, *EMS (Ed)*, p. 604-605, 2006. Avaliable at `http://www.math.arq.uva.es/~edgar/GBLAweb/`.

M. Borges-Quintana, M. Borges-Trenard, P. Fitzpatrick and E. Martínez-Moro. Gröbner bases and combinatorics for binary codes. To appear in Appl. Algebra Engrg. Comm. Comput.

References:

- M. Borges-Quintana, F. Winkler, and M. Borges-Trenard. FGLM Techniques Applied to Linear Codes – An Algorithm for Decoding Linear Codes. *Techn. Rep.*, RISC-Linz, RISC - 00-14, J. Kepler Univ., Linz, Austria, 2000.
- W. D. Brownawell. Bounds for the degrees in the Nullstellensatz. *Ann. of Math.* (2), 126(3), p. 577-591, 1987.
- B. Buchberger. An Algorithmic Criterion for the Solvability of a System of Algebraic Equations (German). *Aequationes Mathematicae* 4, p. 374-383, 1970. (English translation in [1]).
- B. Buchberger, H.M. Möller. The construction of Multivariate Polynomials with Preassigned Zeros. In: EUROCAM'82, Marseille. *LNCS*. 144, p. 24-31, 1982.

References:

- B. Buchberger, F. Winkler (eds). *Gröbner Bases and Applications* (Proc. of the Conference 33 Years of Gröbner Bases). London Mathematical Society Lecture Notes Series 251, C.U.P., 1998.
- J. Faugère, P. Gianni, D. Lazard, T. Mora. Efficient Computation of Zero-Dimensional Gröbner Bases by Change of Ordering. *J. Symb. Comp.* 16(4), p. 329-344, 1993.
- The GAP Group. GAP – Groups, Algorithms, and Programming, Version 4.4.6, 2005. `http://www.gap-system.org`.
- M.G. Marinari, H.M. Möller, T. Mora. Gröbner Bases of Ideals Defined by Functionals with an Application to Ideals of Projective Points. *Appl. Algebra Engrg. Comm. Comput.* 4(2), p. 103-145, 1993.
- T. Mora. *Solving Polynomial Equation Systems II: Macaulay's Paradigm and Gröbner Technology*. Encyclopedia of Maths. and its

Gradient-like decoding of binary linear codes.

M. Borges-Quintana $^{(\star)}$

Joined work with: M. A. Borges-Trenard $^{(\star)}$ Edgar Martínez-Moro $^{(\star\star)}$

$(\star)$ Dpto. Matemática
Universidad de Oriente, Cuba

$(\star\star)$ Dpto. Matemática Aplicada
Universidad de Valladolid, Spain

July 5, 2008, S^{3}CM, Soria, España.

Outline

Outline

Admissible ordering

$\prec$ is admissible on $\langle \mathbf{X} \rangle$:

If it is a total ordering on $\langle X \rangle$ s.t., for all $s, t, u \in \langle X \rangle$:

i) $1 \preceq s$.

ii) $t \prec u : (st \prec su$ and $ts \prec us)$.

Return

Example of a Gröbner basis

$F := \{ x^2 - 1, x^2y - 1 \},$

$p := x^2y^2 - y^2.$

Reducing p module F:

With $x^2y - 1$: $\quad y - y^2 = x^2y^2 - y^2 - (x^2y - 1)y.$

With $x^2 - 1$: $\quad 0 = x^2y^2 - y^2 - (x^2 - 1)y^2.$

$G := \{ x^2 - 1, y - 1 \}$ is a Gröbner basis for Ideal(F).

Return

Example of a Gröbner basis

$F := \{ x^2 - 1, x^2y - 1 \}$,

$p := x^2y^2 - y^2$.

Reducing p module F:

With $x^2y - 1$: $y - y^2 = x^2y^2 - y^2 - (x^2y - 1)y$.

With $x^2 - 1$: $0 = x^2y^2 - y^2 - (x^2 - 1)y^2$.

$G := \{ x^2 - 1, y - 1 \}$ is a **Gröbner basis** for $\mathrm{Ideal}(F)$.

Return

Example of a Gröbner basis

$\mathbf{F} := \{ \mathbf{x}^2 - \mathbf{1}, \mathbf{x}^2\mathbf{y} - \mathbf{1} \},$

$\mathbf{p} := \mathbf{x}^2\mathbf{y}^2 - \mathbf{y}^2.$

Reducing p module F:

With $\mathbf{x}^2\mathbf{y} - \mathbf{1}$: $\mathbf{y} - \mathbf{y}^2 = \mathbf{x}^2\mathbf{y}^2 - \mathbf{y}^2 - (\mathbf{x}^2\mathbf{y} - \mathbf{1})\mathbf{y}.$

With $\mathbf{x}^2 - \mathbf{1}$: $\mathbf{0} = \mathbf{x}^2\mathbf{y}^2 - \mathbf{y}^2 - (\mathbf{x}^2 - \mathbf{1})\mathbf{y}^2.$

$\mathbf{G} := \{ \mathbf{x}^2 - \mathbf{1}, \mathbf{y} - \mathbf{1} \}$ is a **Gröbner basis** for **Ideal(F)**.

Return

Example of a Gröbner basis

$\mathbf{F} := \{\, x^2 - 1,\ x^2y - 1 \,\},$

$\mathbf{p} := x^2y^2 - y^2.$

Reducing p module F:

With $x^2y - 1$: $\quad y - y^2 = x^2y^2 - y^2 - (x^2y - 1)y.$

With $x^2 - 1$: $\quad 0 = x^2y^2 - y^2 - (x^2 - 1)y^2.$

$\mathbf{G} := \{\, x^2 - 1,\ y - 1 \,\}$ is a **Gröbner basis** for **Ideal(F)**.

Return

Definition of Gröbner basis

$I = Ideal(F)$, let $T_{\prec}(I) = \{T(f) \mid f \in I\}$ be the semigroup ideal of the maximal terms of I with respect to (w.r.t.) $\prec$.

G is a Gröbner basis of I w.r.t. $\prec$ if and only if

$$\mathbf{T(I)} = \langle \mathbf{T\{G\}} \rangle.$$

The set of maximal terms of I is generated by the set of maximal terms of G.

Top Back

Definition of Gröbner basis

$I = Ideal(F)$, let $T_{\prec}(I) = \{T(f) \mid f \in I\}$ be the semigroup ideal of the maximal terms of I with respect to (w.r.t.) $\prec$.

G is a Gröbner basis of I w.r.t. $\prec$ if and only if

$$\mathbf{T(I)} = \langle \mathbf{T}\{\mathbf{G}\} \rangle.$$

The set of maximal terms of I is generated by the set of maximal terms of G.

Top Back

Definition of Gröbner basis

$I = Ideal(F)$, let $T_{\prec}(I) = \{T(f) \mid f \in I\}$ be the semigroup ideal of the maximal terms of I with respect to (w.r.t.) $\prec$.

G is a Gröbner basis of I w.r.t. $\prec$ if and only if

$$\mathbf{T(I)} = \langle \mathbf{T}\{\mathbf{G}\}\rangle.$$

The set of maximal terms of I is generated by the set of maximal terms of G.

Top Back

Applications of GB

- ⋆ Algebraic Geometry.
- ⋆ Coding Theory.
- ⋆ Criptography.
- ⋆ Differential Equations.
- ⋆ Integer Programming.
- ⋆ Statistics.

Return

Outline

GBLA: Gröbner bases by linear algebra

GBLA $\equiv$ FGLM techniques (in a more general sense).

Let $<$ be a fixed term ordering on $\langle X \rangle$, I a zero-dimenssional ideal.

$Span_K(N_<(I))$ is represented by

- a K-vector space E with an effective function

  LinearDependency$[v, \{v_1, \ldots, v_r\}]$

  $\{v_1, \ldots, v_r\} \subset E$ linear independent vectors

  $$\begin{cases} \{\lambda_1, \ldots, \lambda_r\} \subset K & \text{si } v = \sum_{i=1}^{r} \lambda_i v_i, \\ \textbf{False} & \text{if } v \text{ is not a linear combination of } \{v_1, \ldots, v_r\}. \end{cases}$$

- an injective morphism $\xi : Span_K(N_<(I)) \mapsto E$.

GBLA: Gröbner bases by linear algebra

GBLA $\equiv$ FGLM techniques (in a more general sense).
Let $<$ be a fixed term ordering on $\langle X \rangle$, I a zero-dimenssional ideal.

$Span_K(N_<(I))$ is represented by STOP

- a K-vector space E with an effective function

  $$\textbf{LinearDependency}[v, \{v_1, \ldots, v_r\}]$$

  $\{v_1, \ldots, v_r\} \subset E$ linear independent vectors

  $$\begin{cases} \{\lambda_1, \ldots, \lambda_r\} \subset K & \text{si } v = \sum_{i=1}^{r} \lambda_i v_i, \\ \textbf{False} & \text{if } v \text{ is not a linear combination of } \{v_1, \ldots, v_r\}. \end{cases}$$

- an injective morphism $\xi : Span_K(N_<(I)) \mapsto E$.

GBLA pattern algorithm

Input: $\prec$, a term ordering on $\langle X \rangle$;
$\xi : Span_K(N_<(I)) \mapsto E$, a suitable representation of $Span_K(N_<(I))$.

Output: $rGb(I, \prec)$.

$<$ could be equal to $\prec$.

Top | Back | More . . .

GBLA algorithm

$G := \emptyset$; $List := \{1\}$; $N := \emptyset$; $r := 0$;
While $List \neq \emptyset$ **do**
  $t :=$ **NextTerm**$[List]$;
  If $t \notin T_{\prec}(G)$ **then**
    $\mathbf{v} := \xi(\mathbf{Can}(\mathbf{t}, \mathbf{I}, <))$; $\Longleftarrow$
    $\mathbf{\Lambda} :=$ **LinearDependency**$[\mathbf{v}, \{\mathbf{v_1}, \ldots, \mathbf{v_r}\}]$; $\Longleftarrow$
    If False $\neq \Lambda$ **then** $G := G \cup \{t - \sum_{i=1}^{r} \lambda_i t_i\}$
     $(where\ \ \Lambda := (\lambda_1, \ldots, \lambda_r))$
    else $r := r + 1$
      $v_r := v$;
      $t_r := t$; $N := N \cup \{t_r\}$;
      $List :=$ **InsertNexts**$[t_r, List]$;
Return$[G]$.

Top Back More ...

GBLA: Main objects

N: A set of representative elements for $K\langle X\rangle/I$.

Matphi (ϕ):

- ⋆ Allows to perform multiplication in N.
- ⋆ To reduce an element in $K\langle X\rangle$ to its representative in N **(Allows to solve the Word Problem)**.

Top Back More . . .

GBLA: Main objects

N: A set of representative elements for $K\langle X\rangle/I$.

Matphi (ϕ):

- Allows to perform multiplication in N.
- To reduce an element in $K\langle X\rangle$ to its representative in N **(Allows to solve the Word Problem)**.

Top Back More . . .

GBLA: Main objects

N: A set of representative elements for $K\langle X\rangle/I$.

Matphi (ϕ):

- $\star$ Allows to perform multiplication in N.
- $\star$ To reduce an element in $K\langle X\rangle$ to its representative in N **(Allows to solve the Word Problem)**.

Top Back More . . .

Other possible outputs

Gröbner representation: (N, ϕ), $N := \{h_1, \ldots, h_s\}$ s.t:

$$K\langle X\rangle / I \cong Span_K(N), \text{ and}$$

Matphi structure:

$$\phi(k): \quad N \times X \longrightarrow K$$

$$\forall_{i \in [1,s]} \quad (h_i x_k = \sum_{j=1}^{s} \phi(k)[h_i, x_j] h_j). \quad \text{(in the quotient)}$$

Border basis: $\mathcal{B}(I, \prec) := \{w - Can(w, I, \prec) \mid w \in B_\prec(I)\}$

$$B_\prec(I) = \{w \mid w \in T(I) \text{ and } \exists\, v \in N \text{ and } x \in X \text{ s.t. } w = vx\}$$

(the border of $T(I)$).

$\mathbf{N = N_\prec(I)}$ if $\prec$ is admissible.

Top Back