Computing invariants of permutation groups using Fourier Transform

Nicolas Borie

March 1, 2011

Short presentation of the problem

- **Data :** Let $\mathbb{C}$ be the complex field. Let n be an integer such that $n \geqslant 1$. Let G be a subgroup of S_n. (i.e. a group of permutations)

Short presentation of the problem

- **Data :** Let $\mathbb{C}$ be the complex field. Let n be an integer such that $n \geqslant 1$. Let G be a subgroup of S_n. (i.e. a group of permutations)
- **Fact :** let $R = \mathbb{C}[x_1, x_2, \dots, x_n]^G$ be the set of polynomials invariant under the action of G. R is a graded connected finitely generated algebra over $\mathbb{C}$. It is also a free module over the symmetric polynomials $\mathbb{C}[x_1, x_2, \dots, x_n]^{S_n}$.

What does we want ?

- **Goal :** find the polynomials in R invariant under the action of G which generates R as an algebra.

What does we want ?

- **Goal :** find the polynomials in R invariant under the action of G which generates R as an algebra.
- Example : The family $\{x_1 + x_2 + x_3, x_1x_2 + x_1x_3 + x_2x_3, x_1x_2x_3\}$ generate $\mathbb{C}[x_1, x_2, x_3]^{s_3}$.

What does we want ?

- **Goal :** find the polynomials in R invariant under the action of G which generates R as an algebra.
- Example : The family $\{x_1 + x_2 + x_3, x_1 x_2 + x_1 x_3 + x_2 x_3, x_1 x_2 x_3\}$ generate $\mathbb{C}[x_1, x_2, x_3]^{S_3}$.
- Find the secondary invariants (polynomials invariants under the action of G but not invariants under S_n).

What does we want ?

- **Goal :** find the polynomials in R invariant under the action of G which generates R as an algebra.
- Example : The family $\{x_1 + x_2 + x_3, x_1x_2 + x_1x_3 + x_2x_3, x_1x_2x_3\}$ generate $\mathbb{C}[x_1, x_2, x_3]^{S_3}$.
- Find the secondary invariants
(polynomials invariants under the action of G but not invariants under S_n).
- $x_1^2x_2 + x_2^2x_3 + x_3^2x_1$ is invariant under $C_3 =< (1, 2, 3) >$ but not under the action of S_3.

What does we want ?

- **Goal :** find the polynomials in R invariant under the action of G which generates R as an algebra.
- Example : The family $\{x_1 + x_2 + x_3, x_1x_2 + x_1x_3 + x_2x_3, x_1x_2x_3\}$ generate $\mathbb{C}[x_1, x_2, x_3]^{s_3}$.
- Find the secondary invariants (polynomials invariants under the action of G but not invariants under S_n).
- $x_1^2x_2 + x_2^2x_3 + x_3^2x_1$ is invariant under $C_3 =< (1, 2, 3) >$ but not under the action of S_3.
- (2009) algorithms and computers can compute it efficiently up to $n = 7$ in characteristic 0.

Using Groëbner basis

Choosing an order on the variables, the usual way to dealt the problem is using Groëbner basis. (an average limit is 7-8 variables...).

- Groëbner basis break the symmetries.

Using Groëbner basis

Choosing an order on the variables, the usual way to dealt the problem is using Groëbner basis. (an average limit is 7-8 variables...).

- Groëbner basis break the symmetries.
- Very heavy cost for products of two polynomials.

$$|G| = 100 \quad (\sum_{i=1}^{100} \alpha_i X^i)(\sum_{j=1}^{100} \beta_j X^j) = \sum_{k=1}^{10000} \dots \tag{1}$$

Using Groëbner basis

Choosing an order on the variables, the usual way to dealt the problem is using Groëbner basis. (an average limit is 7-8 variables...).

- Groëbner basis break the symmetries.
- Very heavy cost for products of two polynomials.

$$|G| = 100 \quad (\sum_{i=1}^{100} \alpha_i X^i)(\sum_{j=1}^{100} \beta_j X^j) = \sum_{k=1}^{10000} \dots \tag{1}$$

- We make calculations in the whole algebra $\mathbb{C}[x_1, x_2, \dots, x_n]$.

Using SAGBI-Groëbner basis

To go further in the computation, we can use an analogue of Groëbner basis for Ideals. With this, we keep the use of symmetries. (an average limit is 7-8-9 variables...)

- SAGBI-Groëbner basis preserves the symmetries.

Using SAGBI-Groëbner basis

To go further in the computation, we can use an analogue of Groëbner basis for Ideals. With this, we keep the use of symmetries. (an average limit is 7-8-9 variables...)

- SAGBI-Groëbner basis preserves the symmetries.
- Relatively heavy cost for products of two polynomials.

Using SAGBI-Groëbner basis

To go further in the computation, we can use an analogue of Groëbner basis for Ideals. With this, we keep the use of symmetries. (an average limit is 7-8-9 variables...)

- SAGBI-Groëbner basis preserves the symmetries.
- Relatively heavy cost for products of two polynomials.
- We make calculations in the algebra $\mathbb{C}[x_1, x_2, \ldots, x_n]^G$

Products in symbolic computation

The regular trick to simplify products in symbolic computation is divided the problem . For univariate polynomials, the Fast Frourier Transform appears today as one of the best method. $(O(n\ log(n)))$

- How put the calculation inside a quotient $\mathbb{C}[X]^G / < (\mathbb{C}[X]^{S_n})^+ >$?

Products in symbolic computation

The regular trick to simplify products in symbolic computation is divided the problem . For univariate polynomials, the Fast Frourier Transform appears today as one of the best method. $(O(n\ log(n)))$

- How put the calculation inside a quotient $\mathbb{C}[X]^G / < (\mathbb{C}[X]^{S_n})^+ >$?
- How many points do we have to set ?

Products in symbolic computation

The regular trick to simplify products in symbolic computation is divided the problem . For univariate polynomials, the Fast Frourier Transform appears today as one of the best method. ($O(n\,log(n))$)

- How put the calculation inside a quotient $\mathbb{C}[X]^G / < (\mathbb{C}[X]^{S_n})^+ >$?
- How many points do we have to set ?
- How choosing evaluation points ?

Goals of a new method

- We want to work in $\mathbb{C}[x_1, x_2, \ldots, x_n]^G / \mathbb{C}[x_1, x_2, \ldots, x_n]^{S_n}$ or a like (the important thing is to get rid of primary invariant)

Goals of a new method

- We want to work in $\mathbb{C}[x_1, x_2, \ldots, x_n]^G / \mathbb{C}[x_1, x_2, \ldots, x_n]^{S_n}$ or a like (the important thing is to get rid of primary invariant)
- A controlled product relatively light. (a fixed cost not heavy...)

Some interesting point

Let ρ a n-th primitive root of unity. Let $A = (1, \rho, \rho^2, \ldots, \rho^{n-1})$ be a point of $\mathbb{C}^n$.

$$\begin{aligned}
\prod_{k=1}^{n}(X - \rho^k) &= X^n - 1 \\
&= (X-\rho)(X-\rho^2)\ldots(X-\rho^n) \\
&= X^n - (\sum_{k=1}^{n}\rho^k)X^{n-1} + \cdots + \prod_{k=1}^{n}\rho^k \\
&= X^n - e_1(1,\rho,\rho^2,\ldots,\rho^{n-1})X^{n-1} + \ldots \\
&\cdots + (-1)^n e_n(1,\rho,\rho^2,\ldots,\rho^{n-1})
\end{aligned}$$

The trick for evaluation

Let ρ be a $n^{th}-$primitive root of unity. Let $e_1, e_2, \ldots, e_n$ be the elementary symmetric functions. We have

$$\begin{aligned} e_1(1, \rho, \rho^2, \ldots, \rho^{n-1}) &= 0 \\ e_2(1, \rho, \rho^2, \ldots, \rho^{n-1}) &= 0 \\ \ldots &= 0 \\ e_{n-1}(1, \rho, \rho^2, \ldots, \rho^{n-1}) &= 0 \\ e_n(1, \rho, \rho^2, \ldots, \rho^{n-1}) &= (-1)^{n+1} \end{aligned}$$

Evaluations points

Let $L = \{\sigma((1, \rho, \rho^2, \dots, \rho^{n-1})) | \sigma \in S_n/G\}$

- It define $\frac{n!}{|G|}$ point as the rank of the module : $\mathbb{C}[x_1, x_2, \dots, x_n]^G / \mathbb{C}[x_1, x_2, \dots, x_n]^{S_n}$

Evaluations points

Let $L = \{\sigma((1, \rho, \rho^2, \ldots, \rho^{n-1}))|\sigma \in S_n/G\}$

- It define $\frac{n!}{|G|}$ point as the rank of the module : $\mathbb{C}[x_1, x_2, \ldots, x_n]^G / \mathbb{C}[x_1, x_2, \ldots, x_n]^{S_n}$
- Symmetric polynomials vanishes of the computations (They are send onto $\mathbb{C}(1, 1, 1, \ldots, 1)$)

Evaluations points

Let $L = \{\sigma((1, \rho, \rho^2, \dots, \rho^{n-1}))|\sigma \in S_n/G\}$

- It define $\frac{n!}{|G|}$ point as the rank of the module : $\mathbb{C}[x_1, x_2, \dots, x_n]^G/\mathbb{C}[x_1, x_2, \dots, x_n]^{S_n}$
- Symmetric polynomials vanishes of the computations (They are send onto $\mathbb{C}(1, 1, 1, \dots, 1)$)
- The product of two polynomials in completely controlled, it is a pointwise product of two vectors of evaluations of size $\frac{n!}{|G|}$ with element in $\mathbb{C}(\rho)$.

Evaluations points

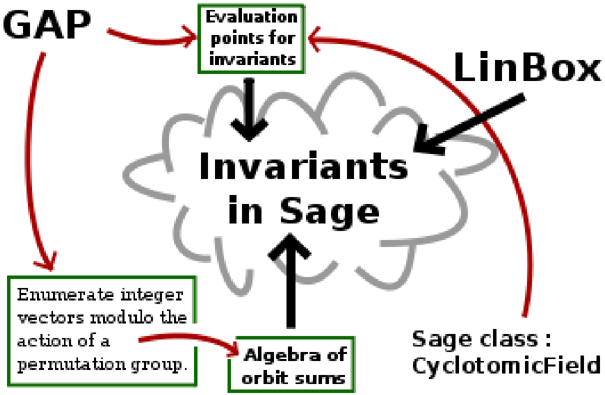
Let $L = \{\sigma((1, \rho, \rho^2, \dots, \rho^{n-1}))|\sigma \in S_n/G\}$

- It define $\frac{n!}{|G|}$ point as the rank of the module : $\mathbb{C}[x_1, x_2, \dots, x_n]^G/\mathbb{C}[x_1, x_2, \dots, x_n]^{S_n}$
- Symmetric polynomials vanishes of the computations (They are send onto $\mathbb{C}(1, 1, 1, \dots, 1)$)
- The product of two polynomials in completely controlled, it is a pointwise product of two vectors of evaluations of size $\frac{n!}{|G|}$ with element in $\mathbb{C}(\rho)$.

- Theorem

  The vectors of evaluation of secondary invariants span $\mathbb{C}^{\frac{n!}{|G|}}$

Implementation in Sage



Benchmark

I really need a standard machine to run my computations and make acceptable comparisons.

Benchmark : TODO

The End.

Thank you.

A powerful system of sharing :
http://www.sagemath.org/

A friendly community :
http://combinat.sagemath.org/