ALGEBRAIC STATISTICS: SOME APPLICATIONS[1]

STICLARU, Gabriel
Faculty of Mathematics-Informatics
Ovidius University of Constanta
gabrielsticlaru@yahoo.com

Abstract

A main purpose of this report is to present the connections between Algebra and Statistics, focused on the applications of Algebra in Statistics. We present applications in experimental designs and we exemplify by using the free computer algebra systems SINGULAR and CoCoA.

Keywords: *Algebraic Statistics, Gröbner bases, Design of Experiments, Computer Algebra System*

AMS Classification: 13P10, 13P25, 62K15

1. **Introduction**

Many people think that Algebra and Statistics have nothing in common, except some applications of Linear Algebra to Statistics. A main purpose of this report is to uncover the numerous connections between these fields, focused on the applications of algebra and its computational tools in Statistics.

Algebraic Statistics is a new field, less than a decade old, whose name was coined by statisticians interested in applying Gröbner bases to the design of experiments (see [8]).

The statistical literature is mainly concerned with explaining techniques but in the last years, abstract algebra have been used to give a mathematical foundation. Areas in algebra include commutative algebra and algebraic geometry, group theory, automata theory, formal languages, combinatorics, graph theory, artificial intelligence, number theory, coding theory and cryptography. It has been applied in design of experiments, hypothesis testing, maximum likelihood estimation, computational biology and finance.

All major computer algebra systems such as SINGULAR, Macaulay2, CoCoA, Axiom, Macsyma, Maple, Mathematica, Reduce contain Gröbner basis algorithms. The most efficient programs are, however, more specialized, and we will mention some free software.

SINGULAR has been developed by a group at the University of Kaiserslautern and is probably the most efficient program for some classes of problems from algebraic geometry and for polynomial computations. We present

[1]Ph.D. Student Paper

some of the most important features of SINGULAR.

Other commonly used is Macaulay, created by D. Bayer and M. Stillman. In addition to Gröbner bases for ideals and modules, many commands are built in, such as ideal operations, calculation of syzygies and Hilbert series.

CoCoA has been developed by a group at the University of Genova. It has a nice interface, and like Macaulay lots of scripts for commutative algebra and algebraic geometry. It also offers lots of strategy choices and different orderings, and allows more general coefficients than Macaulay.

In this report, many computations was made with Computer Algebra Software, SINGULAR and CoCoA.

Many recent work in algebraic statistics has considered polynomials, Grobner bases and toric ideals. Polynomials and ratios of polynomials appear in statistics and probability under various forms, in model representations as well as in inferential procedures. Gröbner bases are a useful tool to obtain a set of different models identified by an experimental design.

In section 3 we introduce the basic algebraic machinery for Statistics: rings of polynomials, ideals, varieties, Gröbner bases and toric ideals.

In algebraic statistics an experimental design is described by a set of polynomials called the design ideal. This is generated by finite sets of polynomials. Two types of generating sets are used in the literature: Gröbner bases and indicator functions. In section 4, we describe them, how they are used in the analysis and planning of a design and how to switch between them. Examples include full factorial designs and fractions of full factorial designs.

2. **Computer Algebra Software**

The growth of algebraic statistics has coincided with the rapid developments of fast symbolic algebra packages such as SINGULAR, CoCoA, and Macaulay 2. The major purpose of a Computer Algebra System (CAS) is to manipulate a formula symbolically using the computer.

ORMS (http://orms.mfo.de) and SIGSAM (www.sigsam.org/software) maintains a collection of references to Mathematical Software project and computer algebra systems.

There are many CAS as commercial systems, with general-purpose (numerical computation, symbolic algebra, functions, graphics, programming): Magma, Maple, MatLab and Mathematica. We give below a short overview of the most important open-source or free general purpose computer algebra software available at the time of writing:

- SINGULAR (www.singular.uni-kl.de): computing SINGULARities.
- CoCoA (http://cocoa.dima.unige.it): Computations in Commutative Algebra.
- Macaulay2 (www.math.uiuc.edu/Macaulay2): algebraic geometry and commutative algebra.
- GAP (www.gap-system.org): Groups, Algorithms and Programming.
- SAGE (http://sage.scipy.org/sage): Software for Algebra and Geometry Experimentation.
- 4ti2 (www.4ti2.de): computation of Hilbert bases, Graver bases, toric Gröbner bases.

- Normaliz (www.math.uos.de/normaliz): computing normalizations.

Let us give a short overview of SINGULAR. There are a lot of documentations about SINGULAR: introductory textbooks, articles, manuals and tutorials (see [5]).

SINGULAR is a computer algebra system for polynomial computations, commutative and non-commutative algebra, algebraic geometry, and singularity theory. It is free and open-source under the GNU license and is available as a binary program (sources C/C++ compiler required), for the most common hard and software platforms:

- Windows 95/98/ME/NT/2K/XP.
- Unix: Linux (PC, DEC-Alpha), HP-UX (Hewlett-Packard), Solaris (Sun), IRIX (SGI), AIX (IBM), OSF (DEC), FreeBSD (PC).
- Macintosh: PPC (need MPW), MacOS X.

SINGULAR's main computational objects are ideals and modules over a large variety of baserings. The baserings are polynomial rings over a field (e.g., finite fields, the rationals, floats, algebraic extensions, transcendental extensions), localizations or quotient rings with respect to an ideal. A general and efficient implementation of communication links allows SINGULAR to make its functionality available to other programs.

SINGULAR is based on other open source software, like GMP and NTL. GMP is a free library for arbitrary precision arithmetic, operating on signed integers, rational numbers, and floating point numbers.

NTL is a high-performance, portable C++ library providing data structures and algorithms for manipulating signed, arbitrary length integers, vectors, matrices, and polynomials.

Because SINGULAR is based on this systems, there is no practical limit to the precision except the ones implied by the available memory in the machine SINGULAR runs on.

Many SINGULAR libraries are for the communication with other software (4ti2, Gfan, Normaliz, Polymake, Topcom), and other software (Macaulay2 and Sage) have interfaces for SINGULAR.

Here are some of the most important features of SINGULAR:

- Large variety of algorithms implemented in kernel(written in C/C++).
- Many algorithms implemented as SINGULAR libraries.
- Computation in many rings, including polynomial rings (SINGULAR is one of the fastest CAS for polynomial computations).
- Ideals Theory (intersection, ideal quotient, elimination and saturation).
- Computations with rational numbers, floating point real numbers.
- A programming language, which is C++ like.

Its advanced algorithms, address topics such as: multivariate polynomial computations, commutative and non commutative homological algebra, invariant theory, solving, linear algebra, singularity theory, deformation theory, normalization, primary decomposition, syzygies and free resolutions of modules, combinatorics, number theory.

SINGULAR comes with a set of standard packages:

- Linear algebra;
- Commutative algebra;
- Algebraic geometry;
- Singularities;
- Invariant theory;
- Symbolic-numerical solving;
- Coding theory;
- System and Control theory;
- Tropical Geometry;
- Non-commutative algebra.

Commutative Algebra package has libraries for computing with polynomials, ideals, algebras and maps, Gröbner bases, toric ideals, homological algebra, integer programming, primary decomposition of modules, normalization, etc.

3. **Computational Commutative Algebra for Statistics**

Many papers (see [4,8]) introduce the use of computational commutative algebra in Design of Experiments (DoE). The conclusions of these papers: "Many models can be solved using methods of commutative algebra and algebraic geometry" means that modern computational algebra packages such as SINGULAR can be used. Algebraic algorithms involve computations in rings like polynomial rings using ideals and Gröbner bases. In last years many innovations have entered from the use of the apparatus of polynomial rings: algebraic varieties, ideals, elimination, quotient operations and so on.

Computational methods in commutative algebra and algebraic geometry has relatively short history. We briefly recall the basic results from commutative algebra we need to develop the subject. For this background, all the computations will be made in SINGULAR and sources for the material in the present section are [1, 3, 6, 7].

The concept of a ring is probably the most basic one in commutative algebra. Best known rings are Z, Q, R, C and the polynomial ring in one or many variables.

In SINGULAR one can define polynomial rings over the following fields:

- the field of rational numbers Q.
- finite fields F_p, p a prime number ≤ 32003.
- finite fields $GF(p^n)$ with p^n elements, p a prime, $p^n \leq\ 215$.
- transcendental extensions of Q or F_p.
- simple algebraic extensions of Q or F_p.
- simple precision real floating point numbers.
- arbitrary prescribed real floating point numbers.
- arbitrary prescribed complex floating point numbers.

Computation in the field of rational numbers

```
LIB "general.lib";
ring Q=0,x,dp;
bigint f=factorial(40);
f; //815915283247897734345611269596115894272000000000
number n = 13579/24680;
n^5; //461677746637035023899/915642150736455680000
bigint bi = binomial(200,100);
bi; //90548514656103281165404177077484163874504589675413336841320
number n;
n=number(30)^40;
n; //12157665459056928801000000000000000000000000000000000000000
```

Let K be a commutative field, and let $R = k[x_1, \ldots, x_n]$ be the polynomial ring over K in the indeterminates $x_1, \ldots, x_n$.

Definitions

A subset $I \subset R$ is an *ideal* if $f + g \in I$ for all $f, g \in I$ and $fg \in I$, for all $f \in I$ and all $g \in R$.

Let F be a set of polynomials in R. The ideal generated by F is denoted by $< F >$ and is given by the set $\{a_1 f_1 + \ldots + a_m f_m \bar{f_i} \in F, a_i \in R\}$.

Let I and J be ideals in R. The sum and product of the two ideals is defined as follows: $I + J = \{f + g : f \in I, g \in I\}$ and $IJ =< F >$, where $F = \{fg : f \in I, g \in J\}$.

The ideal quotient of I by J (colon ideal) is defined as $I : J = \{a \in R : aJ \subset I\}$.

The saturation of I with respect to J is $I : J^\infty = \{a \in R : \exists n$ such that $aJ^n \subset I\}$.

The radical of I, denoted by $\sqrt{I}$ or $\mathrm{rad}(I)$ is the ideal $\sqrt{I} = \{a \in R : \exists n \in N$ such that $a^n \in I\}$.

A standard algebraic construction is the quotient ring R/I for any ideal $I \subseteq R$. The relation $\sim$ defined as $\{f \sim g$ if, and only if, $f - g \in I\}$ is an equivalence relation. The elements of R/I are the equivalence classes. R/I inherits a ring structure from R by defining sum and product of classes as $[f] + [g] = [f + g], [f][g] = [fg]$.

A monomial in n variables (indeterminates) $x_1, \ldots, x_n$ is a power product x^α where $\alpha = (\alpha_1, \ldots, \alpha_n)$, $\alpha_i \in N$. A polynomial in R is a finite K–linear combination of monomials. The presentation of a polynomial as a linear combination of monomials can be unique by choosing a monomial ordering on the set of monomials. A monomial ordering is a total ordering $<$ on the set of monomials (semigroup under multiplication).

A term order is a total order $<$ on the set of all monomials such that:

1. it is multiplicative: $x^a < x^b \Rightarrow x^{a+c} < x^{b+c}$.
2. the constant monomial is the smallest, i.e. $1 < x^\alpha$ for all $\alpha \in N^n \backslash \{0\}$.

The most important monomial orders are listed below. For monomials $u = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ and $v = x_1^{\beta_1} \cdots x_n^{\beta_n}$ one defines:

(lp) the lexicographic order (purely lexicographic) by $u <_{lp} v$ iff for some k one has $\alpha_k < \beta_k$ and $\alpha_i = \beta_i$ for $i < k$;

(dp) the degree reverse lexicographic order by $u <_{dp} v$ iff $\deg(u) < \deg(v)$ or $\deg(u) = \deg(v)$ and for some k one has $\alpha_k > \beta_k$ and $\alpha_i = \beta_i$ for $i > k$.

(Dp) the degree lexicographic order by $u <_{Dp} v$ iff $\deg(u) < \deg(v)$ or $\deg(u) = \deg(v)$ and $u <_{lp} v$.

These three monomial orders satisfy $x_1 > x_2 > \cdots > x_n$. Note that in one variable, there is only one term order: $1 < x < x^2 < x^3 < \ldots <$. For $n = 2$, we have :

- degree lexicographic order: $1 < x_1 < x_2 < x_1^2 < x_1x_2 < x_2^2 < x_1^3 < x_1^2x_2 < \ldots <$
- purely lexicographic order: $1 < x_1 < x_1^2 < x_1^3 < \ldots < x_2 < x_1x_2 < x_1^2x_2 \ldots <$

Some monomial orders

```
ring R1 = 0,(x,y),lp; // ring Q[x,y] and lexicographical order
poly f = x3y + y4 + x4+x2+xy2+xy+y5+x+y; // definition for f
f; //-> x^4+x^3y+x^2+xy^2+xy+x+y^5+y^4+y
ring R2 = 0,(x,y,z),dp;
// ring Q[x,y] and degree reverse lexicographical order
poly f = imap(R1,f); // the same f but in R2
f; // -> y^5+x^4+x^3y+y^4+xy^2+x^2+xy+x+y
ring R3 = 0,(x,y,z),Dp;
//ring Q[x,y] and degree lexicographical order
poly f = imap(R1,f); // the same f but in R3
f; // -> y^5+x^4+x^3y+y^4+xy^2+x^2+xy+x+y
```

Computation in polynomial rings

```
ring R= 0,(x,y),lp; // Q[x,y,z], lp=lexicographical ordering
poly f = 3x3+5y2;
poly g= 2*x^3+4*y^2;
poly h=f-g;
h; // x^3+y^2
ring R = 0,(x,y,z),lp;
poly f = y4z2-x2y2z2+3x6+2z7+3y7+x5+y5+z5;
f; // f in lp order -> 3x^6+x^5-x^2y^2z^2+3y^7+y^5+y^4z^2+2z^7+z^5
leadmonom(f); //leading monomial -> x^6
leadexp(f); //leading exponent for x,y,z:  -> 6,0,0
lead(f); //leading term -> 3x^6
leadcoef(f); //leading coefficient -> 3
```

Factorization (the transformation of the polynomial into a product of polynomials)

```
ring R=0,(x,y),lp;
poly f=x2-y^2;
factorize(f); // factors:  1, x-y, x+y
```

Operations on ideals

```
ring Q=0,(x,y,z),dp;
ideal I=x,y;
ideal J=y2,z;
ideal H=I+J;
H; // -> <x, y, y^2, z>=<x,y,z>
ideal H=intersect(I,J);
H; // -> <y^2, yz, xz>
```

Definitions

Every polynomial $f \in R$ has an initial monomial, denoted by $\mathrm{in}_<(f)$. For every ideal I of R, the initial ideal of I is generated by all initial monomials of polynomials in I: $\mathrm{in}_<(I) = < \mathrm{in}_<(f) : f \in I >$.

A finite subset G of an ideal I is a Gröbner basis (with respect to the term order $<$ if $\{ \mathrm{in}_<(g) | g \in G\}$ generates $\mathrm{in}_<(I)$.

Note: There are many such generating sets. For instance, we can add any element of I to G to get another Gröbner basis.

A reduced Gröbner basis satisfies:

(1) For each g in G, the coeff of $\mathrm{in}_<(g)$ is 1.
(2) The set $\{ \mathrm{in}_<(g) : g \in G \}$ minimally generates $\mathrm{in}_<(I)$.
(3) No trailing term of any g in G lies in the initial ideal $\mathrm{in}_<(I)$.

If F is a set of polynomials, the variety of F over K equals $V(F) = \{(z_1, \ldots, z_n) \in K^n : f(z_1, \ldots, z_n) = 0, f \in F\}$.

Note: The variety depends only on the ideal of F, i.e. $V(F) = V(< F >)$. If G is a Gröbner basis for F, then $V(G) = V(F)$.

A monomial $u = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ is standard if it is not in the initial ideal $\mathrm{in}_<(I)$.

Let be in R^d a finite set $A = \{a_1, \cdots, a_n\} \subset Z^d_{\geq 0}$. Let $K[t] = K[t_1, \cdots, t_d]$ denote the polynomial ring in d variables over a field K. We associate A with the semigroup ring $K[A] = K[t^{\mathbf{a}_1}, \cdots, t^{\mathbf{a}_n}]$, where $t^{\mathbf{a}} = t_1^{a_1} \cdots t_d^{a_d}$ if $a = (a_1, \cdots, a_d)$. Let $K[x] = K[x_1, \cdots, x_n]$ denotes the polynomial ring in n variables over K. The toric ideal I_A of A is the kernel of the surjective homomorphism $\pi : K[x] \longrightarrow K[A]$ defined by setting $\pi(x_i) = t^{\mathbf{a}_i}$ for $1 \leq i \leq n$.

Exemples: If $n = 2$ and $\mathrm{in}_<(I) =< x_1^3, x_2^4 >$, the number of standard monomials is 12. If $\mathrm{in}_<(I) =< x_1^3, x_1x_2^4 >$, then the number of standard monomials is infinite, because x_2^n (n ≥ 1) are standard monomials.

Some basic properties of polynomial rings

R is a factorial domain.

R is Noetherian, so each ideal in R is finitely generated (Hilbert's basis theorem). This means that any ideal I has the form $< F >$ for a finite set of polynomials F.

Fixing an ideal I in R and a term order $<$, there is a unique reduced Gröbner basis for I.

$V(F)$ is empty if and only if $G = \{1\}$ (Hilbert's Nullstellensatz).

The number of standard monomials equals Cardinal($V(I)$), where the zeroes are counted with multiplicity.

The set of standard monomials is a K-basis for the residue ring $\frac{R}{I}$ (i.e., modulo the ideal I, every polynomial f can be written uniquely as a K-linear combination of standard monomials). Given f, there is an algorithm (the division algorithm) that produces this representation (called the normal form of polynomial f) in R.

Note that a Gröbner basis of an ideal I is a particular generator set of I. For every ideal I and term ordering $<$ there exist Gröbner bases of I and a unique reduced Gröbner basis. Gröbner bases of I can be computed from any generator set of I with the Buchberger algorithm which is implemented in most softwares for algebraic computation. For every ideal there is a finite number of reduced Gröbner bases.

Computing Grobner bases

```
// In K[x] with one variable
ring Q=0,(x),dp;
ideal I=x2+3x-4,x3-5x+4;
ideal G=std(I);
G; // < x-1>
Let F = {x^2+xy-10, x^3+xy^2-26, x^4+xy^3-70}.
Here, G = {x-2,y-3} and V(F) = V(G) = {(2,3)}.
ring Q=0,(x,y),dp;
ideal I=x2+xy-10, x3+xy2-26, x4+xy3-70;
ideal G=std(I);
G; // -> <x-2, y-3>
```

Computing toric ideals

```
LIB "toric.lib";
ring r=0,(x1,x2,x3,x4),lp;
intmat A[2][4]=
1,1,1,1,
0,1,2,3;
ideal I=toric_ideal(A,"ect");  // Conti - Traverso algorithm
I;  // -> <x_2x_4-x_3^2, x_1x_4-x_2x_3, x_1x_3-x_2^2>
```

Examples of ideals

For the ring $K[x]$, in one variable, every ideal is principal; that is, is generated by one polynomial, $I =< f >$.

In $R = k[x_1, \ldots, x_n]$: graded ideals, monomial ideals,, square-free monomial ideals, binomial ideals, toric ideals, etc.

4. **Design of Experiments**

In the context of algebraic statistics an experimental design is described by a set of polynomials called the design ideal. This is generated by finite sets of polynomials. We describe them, how they are used in the analysis and planning of a design and how to switch between them. Examples include full factorial designs and fractions of full factorial designs.

We introduce the main concept in *Design of Experiments* (DoE) by an example.

When a company wants to introduce a new product into the market, it is interested to obtain a priory information from the potential customers. Suppose that the strategy of the company is to test five characteristics of the product X_1, X_2, X_3, X_4, X_5 say Color, Shape, Weight, Material and Price and suppose that each variable has three values coded $\{-1, 0, 1\}$. A selected set of potential customers should be asked to rate them on a ordinal scale (from 0 to 10, say). The set of all these points is called a Design. A complete set with all the combinations, is a product set called *Full Design* (or *Full Factorial Design*). Let call $\mathcal{D}$ the set of the $3^5 = 243$ points in our example. For financial or practical reasons, is necessary to determine a subset $\mathcal{F}$ of $\mathcal{D}$, called fraction, from which we can reconstruct a good model.

The polynomial $X(X-1)(X+1) = X^3-X$ vanishes on the set $\{-1, 0, 1\}$. We can say that the polynomial functions X, X^2, X^3 are linearly dependent over $\{-1, 0, 1\}$. Because function X^3 takes the same values as function X we say that these are confounded by $\{-1, 0, 1\}$.

The polynomials which vanishes on $\mathcal{D}$ are called *canonical polynomials* and for our example these are the polynomials $f_i = X_i(X_i-1)(X_i+1)$. All together, they generate an ideal, the defining ideal of $\mathcal{D}$.

4.1. *Design ideal and Gröbner representation*

We consider a design with n factors, where the levels of each factor are coded with rational. A design $\mathcal{F}$ is a finite set of m distinct points in K^m. Let $K[x_1, \cdots, x_m]$ be the polynomial ring of indeterminates $x_1, \cdots, x_m$ with the coefficients in K. The variables in R correspond to the design factors.

The full factorial design of m factors with two levels (2^m design) is expressed as

$$\mathcal{D} = \{(x_1, \cdots, x_m) \mid x_1^2 = \cdots = x_m^2 = 1\} = \{-1, +1\}^m,$$

where we write -1 and 1 as the two levels. We call a subset $\mathcal{F} \subset \mathcal{D}$ a fractional factorial design. Then the set of polynomials vanishing on the points of $\mathcal{F}$

$$I(\mathcal{F}) = \{f \in K[x_1, \cdots, x_m] \mid f(x_1, \cdots, x_m) = 0 \text{ for all } (x_1, \cdots, x_m) \in \mathcal{F}\}$$

is the design ideal of $\mathcal{F}$.

An ideal $I \subset K[x_1, \cdots, x_m]$ is generated by a finite basis $\{g_1, \cdots, g_k\} \subset I$ if for any $f \in I$ there exist polynomials $s_1, \cdots, s_k \in K[x_1, \cdots, x_m]$ such that

$$f(x_1, \cdots, x_m) = \sum_{i=1}^{k} s_i(x_1, \cdots, x_m)g_i(x_1, \ldots, x_m).$$

The above $s_1, \cdots, s_k$ are not unique in general. We write $I = \langle g_1, \cdots, g_k \rangle$ if I is generated by a basis $\{g_1, \cdots, g_k\}$. For example, for the full factorial design of two factors with two levels (2^2-design), the design ideal of $\mathcal{D} = \{-1, +1\}^2$ is written as

$$I(\mathcal{D}) = \langle x_1^2 - 1, x_2^2 - 1 \rangle.$$

Every ideal has a finite basis by the Hilbert basis theorem. In addition, if $\{g_1, \cdots, g_k\}$ is a basis of $I(\mathcal{F})$, then $\mathcal{F}$ coincides with the solutions of the polynomial equations $g_1 = 0, \cdots, g_k = 0$.

Suppose there are n points in a fractional factorial design $\mathcal{F} \subset \mathcal{D}$. A general method to derive a basis of $I(\mathcal{F})$ is to make use of the algorithm for calculating the intersection of the ideals. By definition, the design ideal of the design consisting of a single point, $(a_1, \cdots, a_m) \in \{-1, +1\}^m$, is written as

$$\langle x_1 - a_1, \cdots, x_m - a_m \rangle \subset K[x_1, \cdots, x_m].$$

Therefore the design ideal of the n-points, $\mathcal{F} = \{(a_{i1}, \cdots, a_{im}),\ i = 1, \cdots, n\}$, is given as

$$I(\mathcal{F}) = \bigcap_{i=1}^{n} \langle x_1 - a_{i1}, \cdots, x_m - a_{im} \rangle. \tag{1}$$

To calculate the intersection of ideals, we can use the theory of Gröbner bases. In fact, by introducing the indeterminates $t_1, \ldots, t_n$ and the polynomial ring $K[x_1, \cdots, x_m, t_1, \cdots, t_n]$, is written as

$$I(\mathcal{F}) = I^* \cap K[x_1, \ldots, x_m],$$

where

$$I^* = \langle t_i(x_1 - a_{i1}), \cdots, t_i(x_m - a_{im}),\ i = 1, \cdots, n, t_1 + \cdots + t_n - 1 \rangle \tag{2}$$

is an ideal of $K[x_1, \cdots, x_m, t_1, \cdots, t_n]$. Therefore we can obtain a basis of $I(\mathcal{F})$ as the reduced Gröbner basis of I^* with respect to a term order satisfying $\{t_1, \cdots, t_n\} \succ \{x_1, \cdots, x_m\}$. This argument is known as the elimination theory, one of the important applications of Gröbner bases.

4.2. *Indicator function*

To define the indicator function of $\mathcal{F}$ we must consider $\mathcal{F}$ as a subset of a larger design $\mathcal{D} \subset K^m$. The indicator function f of $\mathcal{F} \subset \mathcal{D}$ is the response function $f(a) = \begin{cases} 1 & \text{if } a \in \mathcal{F} \\ 0 & \text{if } a \in \mathcal{D} \setminus \mathcal{F}. \end{cases}$

When the coordinates of the points in $\mathcal{F}$ and $\mathcal{D}$ are known, the indicator function f can be computed using some form of interpolation formula.

The indicator function f is a real valued polynomial: $\sum_{\alpha \in L} b_\alpha\ X^\alpha(a)$, $a \in \mathcal{D}$.

Example

The indicator function of fractional design $\mathcal{F} = \{(1,0), (-1,0), (0,1), (0,-1)\}$ is $f = -2x_1x_2 + x_1^2 + x_2^2$.

This result is obtained by the following instructions in CoCoA language.

```
- - - CoCoa code fraction -> indicator function
- - - Fraction F ={(1, 0), (-1, 0), (0, 1), (0,-1)}
Use R::=Q[x[1..2]];
Define About()
Return
"Compute indicator function from fraction design
call Fraction2Indicator(Points,D), where:
D:=Tuples([-1,0,1], NumIndets());
Points:= [[1,0],[-1,0],[0,1],[0,-1]];
"
EndDefine;
Define Fraction2Indicator(Points,D);
ND:=Len(D); PA:=NewList(ND,0); P:=NewList(ND);
For H:=1 To Len(Points) Do
For K:=1 To Len(PA) Do
If Points[H]=D[K] Then PA[K]:=1 EndIf;
EndFor;
EndFor;
IdD:=IdealAndSeparatorsOfPoints(D);
For K:=1 To Len(PA) Do
P[K]:=PA[K]*IdD.Separators[K]
EndFor;
F:=Sum(P);
Return F;
EndDefine;
D:=Tuples([-1,0,1], NumIndets());
Points:= [[1,0],[-1,0],[0,1],[0,-1]];
About();
Fraction2Indicator(Points,D);
- - - -The output is:   -2x[1] x[2] + x[1]^2 + x[2]^2
```

When working with polynomial ideals, it is useful to choose a standard form for writing the polynomials. This can be done by choosing a term ordering. For any term ordering for which $x_1 \succ x_2$ the reduced Gröbner basis representation of $I(\mathcal{F})$ is given by the three polynomials g_1, g_2, g_3, where $g_1 = x_1^2 + x_2^2 - 1$, $g_2 = x_2^3 - x_2$ and $g_3 = x_1x_2$. The polynomial g_1 indicates that the points of $\mathcal{F}$ are on the unit circle, g_2 that the factor corresponding to x_2 has three levels $0, \pm 1$ and g_3 that at least one coordinate of each point in $\mathcal{F}$ is zero.

If we fix a Gröbner basis of an ideal $I \subseteq R$, then for every equivalence class $[f] \in R/I$ there exists a unique $f' \in [f]$ written as combination of monomials not divisible by any monomial in $\mathrm{LT}(I)$. The polynomial f' is called the normal form of f and we write $\mathrm{NF}(f)$. Hence, Gröbner bases give a tool to effectively perform sum and products in the quotient ring R/I. Given a design $\mathcal{F}$, the quotient ring $R/I(\mathcal{F})$ is a vector space of dimension equal to the cardinality of $\mathcal{F}$. A monomial basis of $R/I(\mathcal{F})$ can be used as support for a statistical (saturated) regression model as the corresponding information matrix is invertible. A vector space basis of $R/I(\mathcal{F})$ can be determined by using Gröbner bases. The monomials which are not in $\mathrm{LT}(I(\mathcal{F}))$ are linearly independent over the design. Call this set $Est_{\mathcal{F}}$ (set of estimators or standard monomials). They are those monomials which are not divided by any of $\mathrm{LT}(g)$ for all g in a Gröbner basis of $I(\mathcal{F})$.

The leading terms of the Gröbner basis elements of $I(\mathcal{F})$ are $\mathrm{LT}(g_1) = x_1^2$, $\mathrm{LT}(g_2) = x_2^3$ and $\mathrm{LT}(g_3) = x_1x_2$. The four monomials $1, x_1, x_2, x_2^2$ are not divisible by these leading terms, equivalently the first four columns of X below give an invertible matrix:

$$X = \left[\begin{array}{rrrrr|l} 1 & x_1 & x_2 & x_2^2 & x_1^2 & a \\ \hline 1 & 1 & 0 & 0 & 1 & (1,0) \\ 1 & 0 & 1 & 1 & 0 & (0,1) \\ 1 & -1 & 0 & 0 & 1 & (-1,0) \\ 1 & 0 & -1 & 1 & 0 & (0,-1) \end{array}\right]$$

We have the set of estimators $Est_{\mathcal{F}}$ $=\{1, x_1, x_2, x_2^2\}$ and the linear response model (regression model) with all estimators are $\alpha + \beta x_1 + \gamma x_2 + \delta x_2^2$.

4.3. *Changing representation*

Let $\mathbf{f}$ be the indicator function of $\mathcal{F}$ in $\mathcal{D} \subset k^m$, $I(\mathcal{D}) = \langle d_1, \cdots, d_p \rangle$ and $I(\mathcal{F}) = \langle d_1, \cdots, d_p, g_1, \cdots, g_q \rangle$. Note that usually the generator set $\{d_1, \cdots, d_p\}$ is known and has an easy structure, often being $\mathcal{D}$ a full factorial design and hence d_j a polynomial in x_j for $j = 1, \cdots, m$. Then, $I(\mathcal{F}) = \langle d_1, \cdots, d_p, f - 1 \rangle$. This means that once $\mathcal{F}$ is known, a Gröbner basis of $I(\mathcal{F})$ is obtained by applying the Buchberger algorithm to $\{d_1, \ldots, d_p, f - 1\}$.

If we apply this algorithm for $\mathcal{F} = (1,0), (-1,0), (0,1), (0,-1)$; f $= -2\,x_1x_2 + x_1^2 + x_2^2, d_1 = x_1^3 - x_1, d_2 = x_2^3 - x_2$, the reduced Gröbner basis r of $I(\mathcal{F})$ must be $\mathcal{G} = \{g_1, g_1, g_1\}$ where $g_1 = x_1^2 + x_2^2 - 1$, $g_2 = x_2^3 - x_2$ and $g_3 = x_1x_2$.

SINGULAR: Indicator function $->$ Gröbner bases

```
ring R=0,(x1,x2),dp;
poly f1=x1^3-x1;
poly f2=x2^3-x2;
poly f3=-2*x1*x2+x1^2+x2^2-1;
ideal I=f1,f2,f3;
ideal G=std(I);
G;
G[1]=x1*x2
G[2]=x1^2-2*x1*x2+x2^2-1
G[3]=x2^3-x2
```

In algebraic geometry a design $\mathcal{F}$ is seen as a zero-dimensional variety. The focus both in algebraic statistics and in this section switches from the design $\mathcal{F}$ to its ideal $I(\mathcal{F})$. As we saw, the Gröbner representation and the indicator function representation of $\mathcal{F}$ are nothing else than two sets of generators of $I(\mathcal{F})$.

References

1. Cohen, A., *Gröbner bases, an introduction in Some Tapas of Computer Algebra*, "Springer", Berlin, 1-33, 1999.
2. Drton, M., Sturmfels B., Sullivant S., *Lectures on Algebraic Statistics*, "Springer", New York, 157-163, 2009.
3. Froberg, F., *An Introduction to Gröbner Bases*, "John Wiley & Sons", Chichester, 43-60, 1998.
4. Gibilisco, P., Riccomagno, E., Rogantin, M.P., Wynn, H.P., *Algebraic and Geometric methods in statistics*, "Cambridge University Press", 27-47, 175-178, 2010.
5. Greuel, G., Pfister, G., *A Singular Introduction to Commutative Algebra*, "Springer", Berlin/Heidelberg, 2002.
6. Kreuzer, M., Robbiano L., *Computational Commutative Algebra 1*, "Springer", Berlin/Heidelberg, 2000.
7. Kreuzer, M., Robbiano, L., *Computational Commutative Algebra 2*, "Springer", Berlin/Heidelberg, 2005.
8. Pistone, G., Riccomagno, E., Wynn, H., *Algebraic Statistics: Computational Commutative Algebra in Statistics*, "CRC Press", 43-71, 2001.