Degroebnerization and its applications: a new approach for data modelling.

Michela Ceria, Teo Mora and Andrea Visconti

Abstract. We propose a new Computer Algebra-based method to give a polynomial model for data and in particular for the problem of reverse engineering for gene regulatory networks, as an alternative to that proposed by Laubenbacher and Stigler. We show that, in order to give the required model, which is composed by polynomials in normal form with respect to the vanishing ideal of the given data points, it is not necessary to compute Gröbner bases or performing any step of Buchberger reduction with polynomials. It is possible and more efficient to use a model in which polynomials are represented via their evaluation at points and to recover the required polynomials by means of linear algebra and combinatorics.

Gröbner bases' theory, besides being used in the framework of polynomials systems' solving, has found applications also in the data modelling framework.
As an example, in [9], the authors propose a Computer Algebra-based approach to study gene regulatory networks as dynamical systems, starting from measured data, which are essentially states of n given genes in the networks at different times. Any state is considered as a point in $(\mathbb{Z}_p)^n$, where p is a prime. Let $\mathbf{X} = \{P_1, ..., P_N\}$ the set of such points. In order to model their data, they want to construct polynomial functions that are *in normal form*, namely reduced modulo the ideal $I(\mathbf{X})$ whose variety is given by the points. They first compute a separator family for $\mathbf{X}$, so they have one polynomial for each point, say Q_i, $1 \leq i \leq N$, such that for $i, j \in \{1, ..., N\}$, $Q_i(P_j) = 1$ if $i = j$ and $Q_i(P_j) = 0$ otherwise. Then, they reduce modulo a Gröbner basis of the ideal of points $I(\mathbf{X})$ some polynomials they find as linear combinations of separator polynomials. The authors also propose their complexity analysis, stating that

> *In summary, the complexity of the algorithm is*
>
> $$O(n^2N^2) + O((N^3 + N)(\log(p))^2 + N^2n^2) + O(n(N-1))2^{cN+N-1}).$$
>
> *It is quadratic in the number n of variables and exponential in the number N of time points.*

Recently, a new approach in Computer Algebra has been proposed, that is *Degroebnerization* [10, 12], which has been explicitly expressed and endorsed in [11, Vol.3,

40.12,41.15]. Degroebnerization means avoiding Gröbner basis computation and Buchberger's reduction as much as possible, leaving their use to the only cases in which it is really necessary. This is due to the computational complexity of Gröbner bases' computation. Such a new approach consists then in finding new ways to solve practical problems that have been originally solved using Gröbner basis computation and Buchberger's reduction. Usually, the "new ways" that can be found consist in using linear algebra and combinatorial methods. This work places itself in this framework. Indeed, here we show a new, degroebnerized approach to solve the problem proposed in [9].

First of all, we recall that in [3], an efficient algorithm to compute a squarefree version for a separator family associated to a finite set of finite points has been proposed.

Given the finite set of simple points $\mathbf{X} = \{P_1, ..., P_N\} \subset \mathbf{k}^n$, let us denote by $P_i = (a_{1i}, ..., a_{ni})$, $1 \leq i \leq N$, their coordinates and by $I(\mathbf{X})$ the ideal of all polynomials vanishing on $\mathbf{X}$. For $1 \leq i, j \leq N$, let

$$c_{i,j} := \begin{cases} 0, & \text{if } i = j \\ \min\{h : \ 1 \leq h \leq n, \ a_{hi} \neq a_{hj}\} & \text{otherwise} \end{cases}$$

The multiplicative factors for separator polynomials (see [10]) are of the form

$$p_{i,j}^{[c_{i,j}]} = \frac{x_{c_{i,j}} - a_{c_{i,j},j}}{a_{c_{i,j},i} - a_{c_{i,j},j}}.$$

The idea is to avoid repeated factors by keeping track of the reciprocal relations among the points' coordinates, storing them in a rooted tree called *point trie* [3]. It has as many branches (and so as many leaves) as the points; its nodes are labelled with the indices of the points in $\mathbf{X}$, while the edges are labelled by the coordinates of the points, with the additional rule that two points share the same path from the root to level i ($1 \leq i \leq n$) if and only if they share the same coordinates, from the first one to the i-th one. Via this trie we can know how many coordinates are shared by two points and so whether some polynomial already vanishes on some point, thus allowing us to avoid a useless multiplicative factor. The complexity of a round of our procedure is $O(min(N, nr))$.

For applications, the normal form of the separator polynomials modulo the ideal of the given points is needed. Generally, what it is done is to take the set $\mathbf{X}$, compute a separator family $\mathcal{Q}$ for it, find a Gröbner basis $\mathcal{G}$ for the ideal of points $I(\mathbf{X})$ and perform Buchberger reduction on the elements in $\mathcal{Q}$ modulo $\mathcal{G}$.
We show now how to get the same result without the need of computing $\mathcal{G}$, nor of performing any step of Buchberger reduction. Our aim is to use Lundqvist's interpolation approach, taking advantage of point tries and Bar Codes to speed up the computation. Let us consider the following proposition.

Proposition 1 ([10])**.** *Let* $\mathbf{X} = \{P_1, ..., P_N\}$ *be a finite set of simple points,* $I := I(\mathbf{X}) \triangleleft \mathbf{k}[x_1, ..., x_n]$ *the ideal of points and* $\mathsf{N} = \{t_1, ..., t_N\} \subset \mathbf{k}[x_1, ..., x_n]$ *such that* $[\mathsf{N}] = \{[t_1], ..., [t_N]\}$ *is a basis for* $A := \mathbf{k}[x_1, ..., x_n]/I$*. Then, for each* $f \in \mathbf{k}[x_1, ..., x_n]$ *we have*

$$\mathrm{Nf}(f, \mathsf{N}) = (t_1, ..., t_N)(\mathsf{N}[\mathbf{X}]^{-1})^t (f(P_1), ..., f(P_N))^t,$$

where $\mathsf{N}[\mathbf{X}]$ *is the matrix whose rows are the evaluations of* N *at the elements of* $\mathbf{X}$ *and* $Nf(f, \mathsf{N})$ *is the normal form of* f *w.r.t.* $I(\mathbf{X})$*.*

The above Proposition 1 shows that, as soon as we have $\mathbf{X}$ and a basis of the quotient algebra $\mathcal{P}/I(\mathbf{X})$, computing the normal form of a polynomial modulo $I(\mathbf{X})$ is only a matter of *evaluations* and *linear algebra*.
Clearly, if one has a Gröbner basis for $I(\mathbf{X})$ with respect to some term ordering, finding a basis for $\mathcal{P}/I(\mathbf{X})$ is trivial, since one such a basis is the Gröbner escalier of $I(\mathbf{X})$.
Anyway, it is possible to get the lexicographical Gröbner escalier $\mathsf{N}(\mathbf{X})$ of $I(\mathbf{X})$ also in a "purely combinatorial" way, that is, only using comparisons among the coordinates of the points and *without needing to compute any polynomial*.
The first result in this framework has been given by Cerlienco and Mureddu [4, 5, 6], who provided an algorithm to compute $\mathsf{N}(\mathbf{X})$. Such algorithm is iterative on $\mathbf{X}$, but it needs recursion on the variables, leading to a bad complexity: $O\left(n^2N^2\right)$, where N is the number of points and n the number of variables. An improved alternative has been given by the Lex Game [8], which, making a large use of tries, improves the complexity to $O(nN + N\min(N, nr))$, which depends also on r, the maximal number of children of a node in the point trie of $\mathbf{X}$. This algorithm, despite being fast, has a crucial drawback: it is not iterative on the points, thing that for applications to data modelling is a great disadvantage, due to the dynamicity of experiments. In [2], we present a new algorithm (called *Iterative Lex Game*, Iter LG for short) that, employing the point trie and a Bar Code [1] to dynamically store the terms, is more performant than Cerlienco-Mureddu's algorithm but it is iterative on $\mathbf{X}$. In particular, the complexity turns out to be $O(N^2 n \log(N))$ and we think it is the best which can be done, keeping iterativity on the points.
The total complexity of our algorithm is $O(Nmin(N, nr)) + O(N^2 nlog(N)) + (nN)^{O(1)}$.

A Bar Code can be implemented in C using concatenated objects, implemented using a linked list of data structures. We need three different lists, namely one containing the terms, one related to the single bars and one containing the different levels of the Bar Code. In our knowledge, the only function computing the Gröbner escalier associated to a finite set of finite points in a combinatorial way, is the function *nonMonomials* of the library *pointid.lib* in the Software Singular [7], which implements Cerlienco-Mureddu's algorithm. Figure 1 indicates the obtained timings, highlighting the advantages, compared to the aforementioned *nonMonomials* function.

$GF(2^m)$	Points	Coord	Singular	Iter LG
2^4	256	4	1.68s	0.13s
2^4	256	4	5.01s	0.11s
2^5	1024	3	16.77s	0.13s
2^6	4096	3	5m 27.04s	2.18s
2^5	1024	3	23.06s	0.18s
2^6	4096	3	6m 9.29s	2.04s
2^8	65536	3	27h 42m	49m 27s
2^{20}	4096	2	-	37.40s

FIGURE 1. Testing activity for the Iterative Lex Game.

References

[1] Ceria, M., *Bar code: a visual representation for finite set of terms and its applications.*, Mathematics in Computer Science, 14(2), 497-513 (2020), online in 2019 doi:10.1007/s11786-019-00425-4

[2] Ceria M., Mora T. *Combinatorics of ideals of points: a Cerlienco-Mureddu-like approach for an iterative lex game*, available in arxiv as arXiv:1805.09165 [math.AC].

[3] Ceria, M., Mora, T., Visconti, A., *Efficient computation of squarefree separator polynomials*, In International Congress on Mathematical Software (pp. 98-104). Springer, Cham, (2018).

[4] Cerlienco L., Mureddu M., *Algoritmi combinatori per l'interpolazione polinomiale in dimensione* ≥ 2, preprint (1990).

[5] Cerlienco L., Mureddu M., *From algebraic sets to monomial linear bases by means of combinatorial algorithms*, Discrete Math. 139, 73 – 87.

[6] Cerlienco L., Mureddu M., *Multivariate Interpolation and Standard Bases for Macaulay Modules*, J. Algebra 251 (2002), 686 – 726.

[7] Decker, W.; Greuel, G.-M.; Pfister, G.; Schönemann, H.: SINGULAR 4-2-0 — A computer algebra system for polynomial computations. http://www.singular.uni-kl.de (2019).

[8] Felszeghy, B., Ráth, B., Rónyai, L., *The lex game and some applications*, Journal of Symbolic Computation 41(6), 663–681 (2006)

[9] Laubenbacher, R., Stigler, B., *A computational algebra approach to the reverse engineering of gene regulatory networks*, Journal of theoretical biology, 229, 4, 523-537, Academic Press.

[10] Lundqvist, S., *Vector space bases associated to vanishing ideals of points*, Journal of Pure and Applied Algebra 214(4), 309–321 (2010)

[11] Mora T., *Solving Polynomial Equation Systems* 4 Vols., Cambridge University Press, I DOI: 10.1017/*CBO*9780511542831 (2003), II DOI: 10.1017/*CBO*9781107340954 (2005), III DOI: 10.1017/*CBO*9781139015998 (2015), IV DOI:10.1017/*CBO*9781316271902 (2016).

[12] Mourrain B. A New Criterion for Normal Form Algorithms. In: Fossorier M., Imai H., Lin S., Poli A. (eds) Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. AAECC 1999. Lecture Notes in Computer Science, vol 1719. Springer, Berlin, Heidelberg (1999)

Michela Ceria
DMMM
Polytechnic of Bari
Bari, Italy
e-mail: michela.ceria@gmail.com

Teo Mora
Dept. of Mathematics
Universiity of Genoa
Genoa, Italy
e-mail: 5919@unige.it

Andrea Visconti
Dept. of Computer Science
University of Milan
Milan, Italy
e-mail: andrea.visconti@unimi.it