Normaliz: algorithms for rational cones and affine monoids

Winfried Bruns

FB Mathematik/Informatik
Universität Osnabrück

wbruns@uos.de

Davis, September 20, 2010

Affine Monoids

Definition

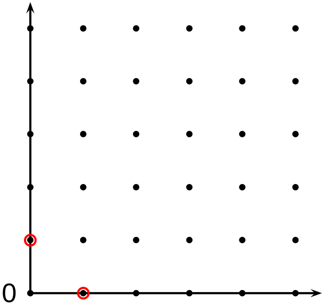
An affine monoid is a finitely generated submonoid of a lattice $\mathbb{Z}^d$, i.e.,

- $M \subset \mathbb{Z}^d$, $M + M \subset M$, $0 \in M$,
- there exist $x_1, \dots, x_n \in M$ such that

$$M = \{a_1 x_1 + \cdots + a_n x_n : a_i \in \mathbb{Z}_+\}.$$

M is positive if $x, -x \in M \Rightarrow x = 0$.

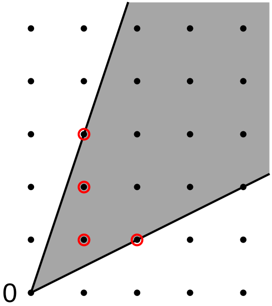
A trivial example



$$M = \mathbb{Z}_+^2$$

(unique) minimal system of generators given by $(1, 0), (0, 1)$

A not so trivial example



$$M = \{x \in \mathbb{Z}^2 : x \leq 2y,\ 3x \geq y\}$$

(unique) minimal system of generators given by

$$(2,1),\ (1,1),\ (1,2),\ (1,3)$$

Cones and lattices

Normaliz computes monoids that arise as intersections of cones and lattices (as the examples above):

Definition

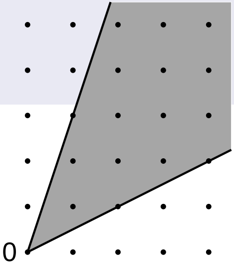
A (rational) cone C is a subset

$$C = \{a_1x_1 + \cdots + a_nx_n : a_1, \ldots, a_n \in \mathbb{R}_+\}$$

with a generating system $x_1, \ldots, x_n \in \mathbb{Z}^d$.

(For us) a lattice is a subgroup of $\mathbb{Z}^d$.

We will often assume $L = \mathbb{Z}^d$.



Gordan's lemma

Theorem

Let $C \subset \mathbb{R}^d$ be the cone generated by $x_1, \dots, x_n \in \mathbb{Z}^d$. Then $C \cap \mathbb{Z}^d$ is an affine monoid.

Proof.

Let $y \in C \cap \mathbb{Z}^d$. Then there exist $a_i \in \mathbb{R}_+$ such that

$$y = a_1 x_1 + \dots + a_n x_n.$$

Write $a_i = b_i + q_i$ mit $b_i \in \mathbb{Z}_+$ and $0 \le q_i < 1$. Then

$$y = b_1 x_1 + \dots + b_n x_n + z, \quad z = q_1 x_1 + \dots + q_n x_n \in C \cap \mathbb{Z}^d.$$

Therefore the monoid $C \cap \mathbb{Z}^d$ is generated by $x_1, \dots, x_n$ and the finite set

$$\mathbb{Z}^d \cap \{q_1 x_1 + \dots + q_n x_n : 0 \le q_i < 1\}$$

□

The Hilbert basis

$x \in M$ is irreducible if $x = y + z \Rightarrow x = 0$ or $y = 0$.

Theorem

Let M be a positive affine monoid.

- *every element of M is a sum of irreducible elements.*
- *M has only finitely many irreducible elements.*
- *The irreducible elements form the unique minimal system of generators $\mathrm{Hilb}(M)$ of M, the Hilbert basis.*

In particular, monoids of type $C \cap \mathbb{Z}^d$ (C pointed, rational) have a unique minimal finite system of generators, often called $\mathrm{Hilb}(C)$.

The task of Normaliz

Normaliz computes (together with other data)

$$\mathrm{Hilb}(C \cap L)$$

Cones C and lattices L can be specified by

- generators $x_1, \dots, x_n \in \mathbb{Z}^d$,
- constraints: homogeneous systems of diophantine linear inequalities, equations and congruences,
- relations: binomial equations.

Normaliz has two algorithms: (1) the original Normaliz algorithm, and (2) a variant of an algorithm due to Pottier.

We concentrate on generators as input and the algorithm (1).

The support hyperplanes

By the theorem of Minkowski-Weyl finitely generated cones can be described by finitely many inequalities:

Theorem

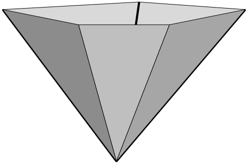
For $C \subset \mathbb{R}^d$ the following are equivalent:

- *there are $x_1, \dots, x_n$ such that $C = \{\sum a_i x_i : a_i \mathbb{R}_+\}$;*
- *there are $\lambda_1, \dots, \lambda_s \in (\mathbb{R}^d)^*$ such that $C = \{x \in \mathbb{R}^d : \lambda_i(x) \geq 0\}$.*

In the case $\dim C = d$ (to which we have restricted ourselves) $\lambda_1, \dots, \lambda_s$ for minimal s define the support hyperplanes

$$H_i = \{x : \lambda_i(x) = 0\}$$

and the facets $F_i = C \cap H_i$ of C.



System of generators and reduction

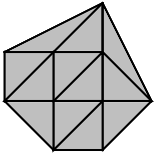
Most likely every algorithm for the computation of Hilbert bases needs two phases:

- the determination of system of generators E,
- the reduction of E to the Hilbert basis.

We need two auxiliary, interleaved steps:

- the computation of the support hyperplanes of the cone,
- a triangulation of the cone.

A triangulation is a decomposition into simplicial cones: $C = \bigcup_{\sigma \in \Sigma} C_\sigma$.



A cone is simplicial if it is generated by linearly independent vectors.

Fourier-Motzkin elimination

This is an incremental algorithm that builds a cone by successive extending the system of generators and determining the support hyperplanes in this process.

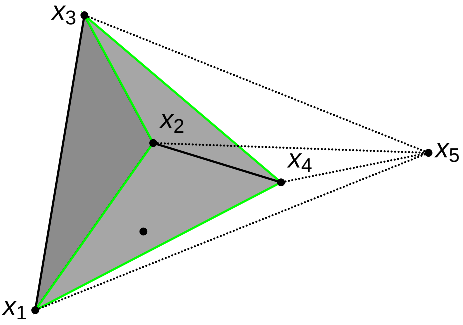
Start: We may assume that $x_1, \dots, x_d$ are linearly independent. The computation of the support hyperplanes is then simply the inversion of the matrix with rows $x_1, \dots, x_d$. (In principle superfluous.)

Extension: we add $x_{d+1}, \dots, x_n$ successively: from the support hyperplanes of $C' = \mathbb{R}_+ x_1 + \cdots + \mathbb{R}_+ x_{n-1}$ we must compute the support hyperplanes of $C = C' + \mathbb{R}_+ x_n$.

We describe this process geometrically.

We determine the boundary V of the part of C' that is visible from x_n and its decomposition into subfacets. Together with x_n these span the new facets of The facets of C' that are visible from x_n $(\lambda_i(x_n) < 0)$, are discarded.

In the cross-section of a 4-dimensional cone:

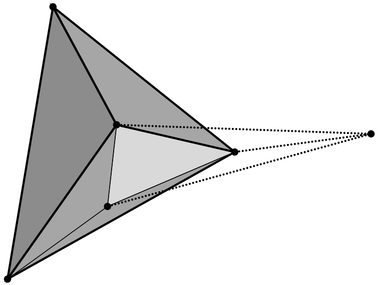


The main problem: find V.

Triangulation

It follows the same inductive scheme, interleaved with Fourier-Moatzkin elimination: we obtain a triangulation of C if we extend the triangulation of C' by the simplicial cones that are spanned by x_n and the facets visible from x_n.

In the cross-section of a 4-dimensional cone:



Main problem: triangulation may be very large. (Way out in certain cases: partial triangulation.)

Reduction

The reduction is in principle very simple if one knows the support hyperplanes (or rather the linear forms $\lambda_1, \dots, \lambda_s$).

Theorem

Let E be a system of generators of the positive normal affine monoid M. An element is $x \in M$ reducible if and only if there exists $y \in E$, $y \neq x$, such that $\lambda_i(x - y) \geq 0$ for $i = 1, \dots, s$.

Evidently true, then for $x, y \in \mathbb{Z}^d$ one has $x - y \in C \cap \mathbb{Z}^d$ if and only if $\lambda_i(x - y) \geq 0$ for $i = 1, \dots, s$.

Main problems:

- E is very large and many comparisons are necessary. In this case a sophisticated implementation can help to find y quickly.
- s is very large.

The steps of the algorithm

The (primal) Normaliz algorithm runs as follows:

1. Input of $x_1, \dots, x_n$, preparatory coordinate transformation.
2. Computation of the support hyperplanes, interleaved with the
3. computation of the triangulation Σ.
4. For every cone $C_\sigma \in \Sigma$
   - computation of a system of generators $C_\sigma \cap \mathbb{Z}^d$ $(*)$ and
   - its reduction to the Hilbert basis HB_σ
5. reduction of $\bigcup_{\sigma\in\Sigma} \mathrm{HB}_\sigma$ to $\mathrm{Hilb}(C \cap \mathbb{Z}^d)$.
6. inverse coordinate transformation.

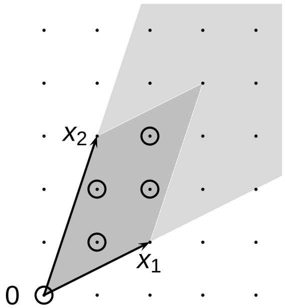
Only step $(*)$ has not yet been explained. Or has it?

Simplicial cones

Let $x_1, \dots, x_d$ be linearly independent and $C = \mathbb{R}_+x_1 + \dots \mathbb{R}_+x_d$. In the proof of Gordan's lemma we have learnt:

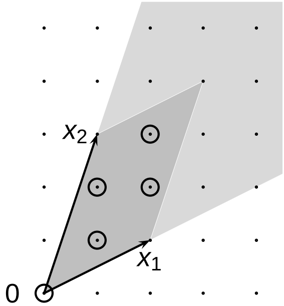
$$E = \{q_1x_1 + \cdots + q_dx_d : 0 \le q_i < 1\} \cap \mathbb{Z}^d$$

together with $x_1, \dots, x_d$ generate the monoid $C \cap \mathbb{Z}^d$.



Easy to see:

Every residue class in $\mathbb{Z}^d/U$, $U = \mathbb{Z}x_1 + \cdots + \mathbb{Z}x_d$, has exactly one representative in U.



Representatives of residue classes can be quickly computed (elementary divisor algorithm) and from an arbitrary representative we obtain the one in E by division with remainder.

Extensions

- Computation of Hilbert series (since 2000, since Version 2.0 based on shellings)
- Pottiers (dual) algorithm (since version 2.1, builds C successively as an intersection of halfspaces)
- parallelization with OpenMP (since version 2.5)
- partial triangulation (since version 2.5)

Monoids from contingency tables

An $r_1 \times r_2 \times \cdots \times r_N$ contingency table is an N-dimensional array $T \in \mathbb{Z}_+^{r_1 \times r_2 \times \cdots \times r_N}$. We consider the marginal distribution

$$\mathscr{M} : T \mapsto (T_1, \dots, T_N) \in \bigoplus_{j=1}^{N} \mathbb{Z}_+^{r_1 \times \dots, \times \widehat{r_j} \times \cdots \times r_N}$$

$$T_j(i_1, \dots, i_{j-1}, i_{j+1}, \dots, i_N) = \sum_{k=1}^{r_j} T(i_1, \dots, i_{j-1}, k, i_{j+1}, \dots, i_N)$$

In the standard case $N = 2$ we just form row and column sums of the matrix T.

Note: Im $\mathscr{M}$ generated by 0-1-vectors with N entries 1 each.

Is Im $\mathscr{M}$ a normal monoid?

Classification by Sullivant and Ohsugi-Hibi left open cases $4 \times 4 \times 3$, $5 \times 4 \times 3$, $5 \times 5 \times 3$.

Challenges mastered

For some monoids arising in algebraic statistics we could compute Hilbert bases:

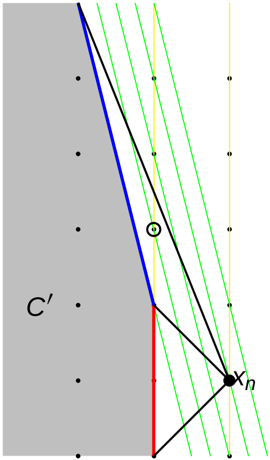
	$4 \times 4 \times 3$	$5 \times 4 \times 3$	$5 \times 5 \times 3$	$6 \times 4 \times 3$	semi-graph-oid $N = 5$
emb-dim	40	47	55	54	32
dim	30	36	43	42	26
# rays	48	60	75	72	80
# HB	48	60	75	4,392	1,300
# supp hyp	4,948	29,387	306,955	153,858	117,978
# full tri	2,654,000	$\approx 10^8$	$\approx 10^{10}$	$\approx 3 * 10^9$	$\approx 10^9$
# partial tri	48	4,320	775,800	206,064	3,109,495
# cand	96	1,260	41,593	10,872	168,014

Most difficult case $5 \times 5 \times 3$, computation time on Intel i7: 50 minutes

Computations also done by R. Hemmecke and M. Köppe (LattE4ti2).

The height 1 strategy

The extremely large examples could only be computed by the height 1 strategy



x_n has height 1 over red facet

$\Rightarrow$ no Hilb candidate

between x_n and red facet

x_n has height 3 over blue facet

$\Rightarrow$ candidates for Hilb

between x_n and blue facet

Team

- Robert Koch (1997-2002, implementation in C)
- Witold Jarnicki (2003)
- Bogdan Ichim (since 2007, cxompletely new implementation in C++, versions 2.0, 2.1, jNormaliz)
- Christof Söger (since 2009, Versions 2.2, 2.5)
- Gesa Kämpf (Macaulay 2 package)
- Andreas Paffenholz (polymake interface)

Sources

- W. Bruns and J. Gubeladze, *Polytopes, rings, and K-theory*. Springer 2009.
- W. Bruns, R. Hemmecke, B. Ichim, M. Köppe, and C. Söger, *Challenging computations of Hilbert bases of cones associated with algebraic statistics*. Preprint, Exp. Math.
- W. Bruns and B. Ichim, *Normaliz: algorithms for affine monoids and rational cones.* Preprint (2009), J. Algebra.
- W. Bruns and G. Kämpf, *A Macaulay 2 interface for Normaliz.* Preprint. J. Softw. Algebr. Geom.
- W. Bruns and R. Koch, *Computing the integral closure of an affine semigroup.* Univ. Iagel. Acta Math. **39** (2001), 59–70.