A polynomial optimisation library in python

Peter Brown

1 Context

Consider a multivariate polynomial $p(x_1, \ldots, x_n)$ such that $p(x_1, \ldots, x_n) \geq 0$ for all $(x_1, \ldots, x_n) \in \mathbb{R}^n)$. Hilbert's 17th problem asked in p can always be written as the sum of squares of rational functions. It was later answered in the positive by Artin [1] and this work laid the foundations for the development of polynomial optimisation.

Asking whether a polynomial is nonnegative is equivalent to asking if the minimum value of the polynomial is below zero. This is a type of optimisation problem known as a polynomial optimisation problem. Such problems take the form

$$\begin{aligned} \max_{x_1,\ldots,x_n} \quad & f(x_1, \ldots, x_n) \\ \text{s.t.} \quad & g_i(x_1, \ldots, x_n) \leq 0 \quad \forall i \end{aligned} \tag{1}$$

where f and g_i are polynomials of the variables $x_1, \ldots, x_n$. Such problems are ubiquitous across science, finding applications in many fields. See, for example, a recent review [2] on their applications to quantum information and computation. Unfortunately, however, they are also NP-Hard to solve [**?**].

Nevertheless, there exist powerful convex optimization techniques that allow us to find convergent approximations to these problems, based on moment matrices [3] and sum of squares polynomials. These techniques can also be extended to cases where the variables do not commute [4].

2 The project

There exists a popular python package for computing approximations to these problems known as ncpol2sdpa [5], which is currently maintained by the project supervisor.[1] However, the package cannot be easily updated to incorporate new theoretical advances in polynomial optimization and advanced polynomial methods to simplify the computations.

Together in this project, our goal will be to build ncpol3sdpa, which will be a flexible and efficient polynomial optimisation python package that users of ncpol2sdpa can easily migrate to.

2.1 Learning goals

By the end of this project, participants should be able to:

- Understand, build and code moment matrix and sum of squares hierarchies for (noncommutative) polynomial optimization.
- Apply polynomial optimisation problems to solve applications in other domains.
- Gain understanding of new advances in the field of polynomial optimisation hierarchies.

2.2 Prerequisites

- Good knowledge and willingness to code in python.
- Interest in the theory of polynomials and optimisation.

[1] See this github fork `https://github.com/peterjbrown519/ncpol2sdpa`.

References

[1] E. Artin, "Über die zerlegung definiter funktionen in quadrate," in *Collected Papers*, pp. 273–288, Springer, 1965.

[2] A. Tavakoli, A. Pozas-Kerstjens, P. Brown, and M. Araújo, "Semidefinite programming relaxations for quantum correlations," *Reviews of Modern Physics*, vol. 96, no. 4, p. 045006, 2024.

[3] J. B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM Journal on optimization*, vol. 11, no. 3, pp. 796–817, 2001.

[4] S. Pironio, M. Navascués, and A. Acín, "Convergent relaxations of polynomial optimization problems with noncommuting variables," *SIAM Journal on Optimization*, vol. 20, no. 5, pp. 2157–2180, 2010.

[5] P. Wittek, "Algorithm 950: Ncpol2sdpa—sparse semidefinite programming relaxations for polynomial optimization problems of noncommuting variables," *ACM Transactions on Mathematical Software (TOMS)*, vol. 41, no. 3, pp. 1–12, 2015.