

HONORS PROJECT

YIXUAN ZHOU AND MAREIKE DRESSLER

ABSTRACT. Short description

1. INTRODUCTION

Polynomial with real coefficients is a very powerful tool in terms of expressing problems. It has a vast range of applications, in both theory side and engineering side. Deciding the nonnegativity of a multivariate polynomial, naturally, become a central problem for many optimizations and feasibility problems. However, it is known that deciding the nonnegativity of an arbitrary multivariate polynomial is a NP-hard problem when the degree of the polynomial is greater than or equals to 2. Therefore, we restrict our focus to decide whether the polynomial can be expressed in *sum of squares (SOS)* form.

Definition 1.1. Sum of Squares (SOS)

Let $\mathbb{R}[x]_{n,d}$ denotes the set of polynomials with n real variables with degree at most d . A polynomial $p(x) \in \mathbb{R}[x]_{n,2d}$, is a *sum of squares* if there exists $q_1(x), \dots, q_n(x) \in \mathbb{R}[x]_{n,d}$ such that

$$(1.1) \quad p(x) = \sum_{i=1}^n q_i^2(x)$$

It is clear that if a polynomial $p(x)$ is a SOS, then it is nonnegative, Thus, SOS is a (proper) subset of the set of nonnegative polynomials.

We are interested in SOS because given a multivariate polynomial, the decision problem of whether it can be decomposed into SOS is a polynomial time problem, by using *Semidefinite Programming (SDP)*.

Theorem 1.2. *A multivariate polynomial $p(x) \in \mathbb{R}[x]_{n,2d}$ is a sum of squares if and only if there exists $\mathcal{Q} \in \mathcal{S}^{\binom{n+d}{d}}$ such that*

$$(1.2) \quad p(x) = [x]_d^T \mathcal{Q} [x]_d \quad \mathcal{Q} \succcurlyeq 0$$

Where $[x]_d$ denotes the vector of monomials with degree at most d .

[BPT13]

As a consequence of the above theorem, whether a multivariate polynomial is SOS can be determined by a *Semidefinite Programming*. Notice that, the size of the semidefinite matrix, $\binom{n+d}{d}$, when fixing d , grows in polynomial time with respect to n , and when fixing

Key words and phrases. some keywords go here.

n , it grows in polynomial time with respect to d . Thus, though imposed some limitation, we have "reduced" a NP-Problem to a P problem.

In the theorem, there is nothing special with monomials other than being a basis of the vector space $\mathbb{R}[x]_{n,2d}$. Instead of using $[x]_d$ to proceed the calculation, we can choose any basis of the vector space of polynomials $\mathbb{R}[x]_{n,d}$. For example, we can choose Lagrange basis, Chebychev basis, ... And in the most cases, the monomial basis is not the right choice due to its instability of monomials.

In this paper, we will use python program to analyze the computational efficiency and numerical stability of the usage of different bases. In particular, the analysis will be focusing on the condition number of the linear system generated when solving the semidefinite program. We further attempt to identify some most effective bases for some particular type of polynomials, and will try to justify the reasons.

2. PRELIMINARIES

Here background definitions etc will be put. YOU can do it in several subsections (like notation, bla, blabla)

3. MAIN STUFF

Proposition 3.1.

Proof.

□

maybe a theorem

Theorem 3.2.

or an example...

Example 3.3.

pictures are always a good idea...

4. MAYBE ANOTHER SECTION

5. RESUME, OUTLOOK, OR/AND OPEN PROBLEMS

what did you do, what questions are still open, natural next steps etc.

REFERENCES

- [BPT13] G. Blekherman, P.A. Parrilo, and R.R. Thomas, *Semidefinite optimization and convex algebraic geometry*, MOS-SIAM Series on Optimization, vol. 13, SIAM and the Mathematical Optimization Society, Philadelphia, 2013.

8514 VILLA LA JOLLA DRIVE # 114, LA JOLLA, CA, 92037

Email address: yiz044@ucsd.edu

9500 GILMAM DRIVE # 0112, LA JOLLA, CA, 92093

Email address: mdressler@ucsd.edu