

HONORS PROJECT

YIXUAN ZHOU AND MAREIKE DRESSLER

ABSTRACT. Short description

1. INTRODUCTION

Polynomials are very powerful tools in solving many problems in applied math and engineering. And deciding the nonnegativity of a given polynomial become very important in many settings (polynomial optimizations and statistical regressions). Therefore, this decision problem become a central topic in the study of real algebraic geometry. In computational complexity terminology, it has been known that the deciding whether an arbitrary multivariate polynomial is a *nonnegative polynomial* 2.10 is a NP-hard problem (computationally infeasible) when the degree of the polynomial is greater than two. However, if we redirect our focus to determine whether a given polynomial can be expressed in terms of *sum of squares (SOS)* 2.12, the problem is reduced to solving a *semidefinite programming (SDP)* 2.18 problem which has known algorithms that can solve the problem in polynomial times (efficiently) with respect to the input size.

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)*.

[BPT13]

[Lau09]

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 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 above 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 bases.

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

Key words and phrases. some keywords go here.

semidefinite program, that is generated by $p(x) = [x]_d^T \mathcal{Q}[x]_d$. We will use the *condition number* of the matrix in the linear system to identify how stable the it is under noises that are introduced in practical problems. We will 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.

2.1. Linear Algebra And Semidefinite Programming.

2.2. Real Coefficient Polynomials. Because we are going to use a lot of tools from linear algebra, we first introduce some key definitions that we will use in this thesis.

Definition 2.1. Given a matrix $A \in \mathbb{R}^{n \times n}$, we say it is *symmetric* if $A^T = A$.

Definition 2.2.

Theorem 2.3. Spectrum Theorem

Given a symmetric matrix $A \in \mathbb{R}^{n \times n}$, it can be diagonalized as

$$(2.1) \quad A = P^{-1}DP$$

where D is the diagonal matrix with all real values, and P is an orthonormal matrix.

In other word, A has all real eigenvalues, and their corresponding eigenvectors form an orthonormal basis of \mathbb{R}^n . [Gol96]

Definition 2.4. Given a matrix $A \in \mathbb{R}^{n \times n}$, it is *positive semidefinite* if A is symmetric and

$$(2.2) \quad x^T Ax \geq 0 \quad \forall x \in \mathbb{R}^n$$

And we denote it as $A \succcurlyeq 0$.

Proposition 2.5. A matrix is positive semidefinite if and only if all its eigenvalues are greater than or equals to 0

Proof. If a matrix A has eigenvalue $\lambda < 0$, then if x is the corresponding eigenvector, we have $x^T Ax = \lambda x^T x < 0$. On the other hand, if A has all positive eigenvalues, because A being a symmetric matrix, its eigenvectors form a basis. Thus for any $x \in \mathbb{R}^n$, we have $x = \sum_{i=1}^n c_i v_i$ where v_i are the eigenvectors of A , that are also orthonormal to each other. Hence, $x^T Ax = \sum_{i=1}^n \lambda_i c_i^2$. Since all $\lambda_i \geq 0$, we have that $x^T Ax \geq 0$. \square

Definition 2.6. Given a matrix $A \in \mathbb{R}^{n \times m}$, the *pseudo-inverse*, which is also known as the *Moore-Penrose* inverse of A , is the matrix A^+ satisfying:

- $AA^+A = A$
- $A^+AA^+ = A^+$
- $(AA^+)^T = AA^+$
- $(A^+A)^T = A^+A$

Every matrix has its pseudo-inverse, and when $A \in \mathbb{R}^{n \times m}$ is *full rank*, that is $\text{rank}(A) = \min\{n, m\}$, A can be expressed in simple algebraic form.

In particular, when A has linearly independent columns, A^+ can be computed as

$$(2.3) \quad A^+ = (A^T A)^{-1} A^T$$

In this case, the pseudo-inverse is called the *left inverse* since $A^+A = I$.

And when A has linearly independent rows, A^+ can be computed as

$$(2.4) \quad A^+ = A^T(AA^T)^{-1}$$

In this case, the pseudo-inverse is called the *right inverse* since $AA^+ = I$.

Definition 2.7. Given a matrix $A \in \mathbb{R}^{n \times m}$, the condition number of A , $\kappa(A)$ is defined as

$$(2.5) \quad \kappa(A) = \|A\| \cdot \|A^+\|$$

for any norm imposed on A , for instance *Frobenius norm*.

Remark 2.8. The condition number measure how stable the system is. It can be alternatively defined as

$$(2.6) \quad \kappa(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$$

where the σ denotes the singular values of A .

Thus, it can be understood as how stable our system is. Intuitively, when the condition number is large, some error in the input along the max direction of the singular value, our result would largely fluctuate because the error, magnified by the singular value, will dominate the input that is along the direction of the minimum singular value. Therefore, the smaller the condition number is, the more stable our system is under fluctuations caused by noises. The rigorous explanation of the condition number can be found in [CK07]

Now with the tools from linear algebra, we are ready to proceed to the realm of real coefficient polynomials.

Definition 2.9. Let $\mathbb{R}[x]_{n,d}$ denotes the set of real coefficient polynomials with n variables and at most d degree.

Definition 2.10. Let $P_{n,2d}$ denotes the set of nonnegative polynomials with n variables and at most $2d$ degree, that is

$$(2.7) \quad P_{n,2d} = \{p \in \mathbb{R}[x]_{n,2d} : p(x) \geq 0, \forall x \in \mathbb{R}^d\}$$

Remark 2.11. When trying to determine the nonnegativity of a polynomial, there is no reason to consider the set $P_{n,d}$ when d is odd, since if the degree of a polynomial is odd, then it will always be nonnegative at some point.

Definition 2.12. Let $\Sigma_{n,2d}$ denotes the set of polynomials with n variables and at most d degree that are *Sum of Squares*, that is

$$(2.8) \quad \Sigma_{n,2d} = \{p \in \mathbb{R}[x]_{n,2d} : \exists q_1(x), \dots, q_k(x) \in \mathbb{R}[x]_{n,d} \text{ s.t. } p(x) = \sum_{i=1}^k q_i^2(x)\}$$

Remark 2.13. It is easy to check that $P_{n,2d}$ form an vector space. There are a lot of choices of basis, and the most canonical one is the monomial basis.

Example 2.14. $P(2,2)$ is a vector space, then $\mathcal{B}_{2,2} = x^2, xy, y^2, x, y, 1$ is a basis of the vector space.

Remark 2.15. Given $\mathcal{B}_{n,d}$ be a basis of $P_{n,d}$. If we list the elements of $\mathcal{B}_{n,d}$ in a column vector, write as b , then bb^T form a matrix whose upper triangle entries can be collected to form a basis of $P_{n,2d}$.

Example 2.16. Let $\mathcal{B}_{2,1} = x, y, 1$, be a basis of $P_{2,1}$, then

$$(2.9) \quad \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} x^2 & xy & x \\ xy & y^2 & y \\ x & y & 1 \end{bmatrix}$$

Then the upper triangle of the result form a basis of $P_{2,2}$.

Then the question that we are interested in is given any polynomial $p(x) \in P_{n,2d}$, decide whether it is in $\Sigma_{n,2d}$. To help solving this decision problem, the following proposition will become very helpful.

Proposition 2.17. Given $p(x) \in P_{n,2d}$, if $p(x) \in \Sigma_{n,2d}$, then for any basis $\mathcal{B}_{n,d}$ of $P_{n,d}$, there exists a matrix $\mathcal{Q} \succcurlyeq 0$ such that

$$(2.10) \quad \mathcal{B}_{n,d}^T \mathcal{Q} \mathcal{B}_{n,d} = p(x)$$

Proof. For any $p(x) \in P_{n,2d}$, if $p(x) \in \Sigma_{n,2d}$, then we can write

$$(2.11) \quad p(x) = \sum_{i=1}^k q_i^2(x) = [q_1(x), \dots, q_k(x)] \begin{bmatrix} q_1 \\ \vdots \\ q_k \end{bmatrix}$$

Notice that $q_j(x) \in P_{n,d}$.

Now given $\mathcal{B}_{n,d} = \{b_1, \dots, b_{\binom{n+d}{d}}\}$ be a basis of $P_{n,d}$, we have

$$(2.12) \quad q_j(x) = \sum_{i=1}^{\binom{n+d}{d}} c_{ij} b_i = [c_{1j}, \dots, c_{\binom{n+d}{d}j}] \begin{bmatrix} b_1 \\ \vdots \\ b_{\binom{n+d}{d}} \end{bmatrix}$$

By substituting the section equation into the first, we have

$$(2.13) \quad p(x) = \begin{bmatrix} b_1 & \dots & b_{\binom{n+d}{d}} \end{bmatrix} \begin{bmatrix} c_{1,1} & \dots & c_{1,k} \\ \vdots & & \\ c_{\binom{n+d}{d},1} & \dots & c_{\binom{n+d}{d},k} \end{bmatrix} \begin{bmatrix} c_{1,1} & \dots & c_{1,\binom{n+d}{d}} \\ \vdots & & \\ c_{k,1} & \dots & c_{k,\binom{n+d}{d}} \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_{\binom{n+d}{d}} \end{bmatrix}$$

Now the matrices in the middle is $C^T C = \mathcal{Q}$ a positive semidefinite matrix, which proves the forward direction of this theorem.

On the other hand, if we know $p(x) = \mathcal{B}_{n,d}^T \mathcal{Q} \mathcal{B}_{n,d}$, where \mathcal{Q} is a positive semidefinite matrix, we can just apply the Cholesky decomposition to get $\mathcal{Q} = L^T L$, and recover the SOS form of $p(x)$. [BPT13] \square

Therefore, we have reduced our problem decision problem to finding this positive semidefinite matrix. It turns out that this can be done via *Semidefinite Programming* [BPT13]

Definition 2.18. An *Semidefinite Problem (SDP)* in standard primal form is written as

$$(2.14) \quad \text{minimize } \langle C, X \rangle \quad \text{subject to } \langle A_i, X \rangle = b_i, i = 1, \dots, k \quad X \succcurlyeq 0$$

Returning to our problem, we require $Q \succcurlyeq 0$, and the set of constraints $\langle A_i, X \rangle = b_i$ is the same as satisfying $p(x) = \mathcal{B}_{n,d}^T Q \mathcal{B}_{n,d}$, which can be done via comparing the coefficients of each element in the basis that we use. [Rec14] Therefore, when we plug in the constraints into a SDP solver, if the result that we get is not no solution, then we know that the $p(x) \in \Sigma_{n,2d}$

Example 2.19. We can do an example here

Now, we further examine the constraints. Using the inner product of matrices, there is a natural representation of the constraints using the inner products $\langle \cdot, \cdot \rangle$ between matrices. That is $\langle A, B \rangle = \text{tr}(A^T B)$

Proposition 2.20. We pick a basis of $\mathcal{B}_{n,d} = \{b_1, \dots, b_{\binom{n+d}{d}}\}$ of $P_{n,d}$, and list it in a vector form $b = \begin{bmatrix} b_1 & \dots & b_{\binom{n+d}{d}} \end{bmatrix}^T$. Then by remark 2.14, we can form a basis $\mathcal{B}_{n,2d} = \{b'_1, \dots, b'_{\binom{n+2d}{2d}}\}$ based on the vector b . Suppose the $p(x)$ that we are interested in is written in the form $\sum_{i=1}^{\binom{n+2d}{2d}} c_i b'_i$. Then, we can reformulate $p(x) = b^T Q b = \langle Q, b \dots b^T \rangle$

Because the constraints are linear with respect to Q , we can re-write it into a system on linear equations with respect to the entries of the matrix Q . Thus we can reformulate the constraints as $Aq = b$, where $q = [q_{1,1}, \dots, q_{1,k}, q_{2,2}, \dots, q_{k,k}]$. And we are interested in the numerical stability of A .

The stability of A can be measured by the *condition number*. Therefore, we shall seek for the base that minimize $\kappa(A)$. [Rec14]

Definition 2.21. We call the matrix $b \cdot b^T$ in Proposition 2.19 the *Moment Matrix*.

Example 2.22. Here is another place to do an example to illustrate Moment Matrix.

Utilizing the *Moment Matrix*, we can already measure the numerical stability for the different bases $\mathcal{B}_{n,d}$ that one can choose to carry out the comparing coefficient process to generate the constraints in the *SDP*. However, what if the polynomial $p(x)$ used a different base comparing to the *moment matrix*? Given the polynomial $p(x)$ in a basis $\mathcal{B}'_{n,2d}$, one can find a change of basis matrix to convert write $p(x)$ in $\mathcal{B}_{n,2d}$ that we will use to compare coefficients. This change of coefficients process has its own numerical properties as well. So it would be desired if we can take that into account when analyzing the numerical properties of our constraints. Therefore, we shall introduce the *Coefficient Moment Matrix*. In the remaining part of this section, we shall build our way to it.

Suppose the *moment matrix* is constructed with the basis $\mathcal{B}_{n,d}$ and the polynomial $p(x)$ is written in the basis $\mathcal{B}'_{n,2d}$. That is suppose $\mathcal{B}'_{n,2d} = \{b'_0, \dots, b'_k\}$ where $k = \binom{n+2d}{2d} - 1$, we have $p(x) = c_0 b'_0 + \dots + c_k b'_k$.

Definition 2.23. We define the *coefficient extraction map* as the following map,

$$(2.15) \quad \begin{aligned} \mathcal{C} : \mathbb{R}[x]_{\leq, 2d} \times \mathbb{R}[x]_{\leq, 2d} &\rightarrow \mathbb{R} \\ \mathcal{C}(p, b'_j) &\mapsto c_j \end{aligned}$$

[Rec14]

When fixing s_j , we have the *coefficient extraction map* being a linear map with respect to the polynomial $p(x)$. Indeed, we have

$$(2.16) \quad \begin{aligned} \mathcal{C}(\lambda p, b'_j) &= \lambda \mathcal{C}(p, b'_j) \quad \lambda \in \mathbb{R} \\ \mathcal{C}(p_1 + p_2, b'_j) &= \mathcal{C}(p_1, b'_j) + \mathcal{C}(p_2, b'_j) \end{aligned}$$

Remark 2.24. When the $\mathcal{B}'_{n, 2d} = \{b'_1, \dots, b'_k\}$ is a orthonormal basis, i.e. $\langle b_i, b_j \rangle \delta_{i,j}$ (the Dirac delta function), where the inner product is defined as

$$(2.17) \quad \langle p, q \rangle = \int_a^b p(x)q(x)d\alpha(x)$$

. There is a natural concretely definition for the coefficients extraction map. That is

$$(2.18) \quad \mathcal{C}(p(x), b'_j) = \langle p(x), b'_j \rangle$$

When the basis is only orthognal, we can still define the coefficients extration map concretely as,

$$(2.19) \quad \mathcal{C}(p(x), b'_j) = \frac{1}{\|b'_j\|} \langle p(x), b'_j \rangle$$

where the norm $\|\cdot\|$ is induced by the corresponding inner product.

Remark 2.25. We should actually write the coefficient extraction map as $\mathcal{C}_{\mathcal{B}'_{n, 2d}}$, since it depends on the base itself. However, when the base is clear, we just write it as \mathcal{C} .

We can generalize the *coefficients extraction map* to take in a matrix as the first argument, and just entry-wise apply the map. With an abuse of notation, we have

Definition 2.26. Given a matrix of polyomials $(p_{i,j}(x))_{i,j}$ all in the bases Let the *coefficient extration map* be defined as

$$(2.20) \quad \mathcal{C} \left(\begin{bmatrix} p_{1,1} & \dots & p_{1,n} \\ \vdots & & \vdots \\ p_{m,1} & \dots & p_{m,n} \end{bmatrix}, b'_j \right) = \begin{bmatrix} \mathcal{C}(p_{1,1}, b'_j) & \dots & \mathcal{C}(p_{1,n}, b'_j) \\ \vdots & & \vdots \\ \mathcal{C}(p_{m,1}, b'_j) & \dots & \mathcal{C}(p_{m,n}, b'_j) \end{bmatrix}$$

Remark 2.27. And immediate result from the above definition is that, given $Q \in \mathbb{R}^{m \times m}$, $M \in \mathbb{R}[x]_{n, 2d}^{m \times m}$, and a basis $\mathcal{B}'_{n, 2d} = \{b'_1, \dots, b'_k\}$, the matrix inner product provides the following relation,

$$(2.21) \quad \mathcal{C}(\langle Q, M \rangle, b'_j) = \langle Q, \mathcal{C}(M, b'_j) \rangle$$

Notice that, let $\mathcal{B}_{n,d} = \{b_0, \dots, b_l\}$, where $l = \binom{n+d}{d} - 1$, let $\vec{b} = [b_1, \dots, b_l]^T$, $M = \vec{b} \cdot \vec{b}^T \in \mathbb{R}^{l,l}$. Then given $p(x)$ in $\mathcal{B}'_{n,2d} = \{b'_0, \dots, b'_k\}$, $p = c_0 b'_0 + \dots + c_k b'_k$, given $Q \in \mathbb{R}^{l,l}$ be the change of basis matrix from $\mathcal{B}_{n,2d}$ to $\mathcal{B}'_{n,2d}$, where $\mathcal{B}_{n,2d}$ is generated by $\mathcal{B}_{n,d}$ using remark 2.14, we have

$$(2.22) \quad c_j = \mathcal{C}(p, b'_j) = \mathcal{C}(\langle Q, M \rangle, b'_j) = \langle Q, \mathcal{C}(M, b'_j) \rangle$$

Definition 2.28. Define the matrix $\mathcal{A}_j = \mathcal{C}(M, b'_j)$ be the *coefficient moment matrix* of b'_j .

Proposition 2.29. \mathcal{A}_j is symmetric, because M is symmetric.

Recall, the *SOS* problem is to decide, given a polynomial $p(x)$, whether there exists a $Q \preceq 0$ such that $p = \vec{b}^T Q \vec{b}$, where \vec{b} be the vector generated by $\mathcal{B}_{n,d}$. Suppose $p(x)$ is given in $\mathcal{B}'_{n,2d}$, we can then reformulate the constraints $\vec{b}^T Q \vec{b}$ using the *coefficient moment matrix* as

$$(2.23) \quad \langle Q, \mathcal{A}_j \rangle = c_j \quad \forall j = 0, \dots, \binom{n+2d}{2d} - 1$$

$$\text{Let } \mathcal{A}_j = \begin{bmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,l} \\ a_{0,1} & a_{1,1} & \dots & a_{1,l} \\ & \vdots & & \\ a_{0,l} & a_{1,l} & \dots & a_{l,l} \end{bmatrix}, Q = \begin{bmatrix} q_{0,0} & q_{0,1} & \dots & q_{0,l} \\ q_{0,1} & q_{1,1} & \dots & q_{1,l} \\ & \vdots & & \\ q_{0,l} & q_{1,l} & \dots & q_{l,l} \end{bmatrix}$$

Set $\vec{a}_j = [a_{0,0}, 2a_{0,1}, \dots, 2a_{0,l}, a_{1,1}, 2a_{1,2}, \dots, a_{l,l}]^T \in \mathbb{R}^{l+1}$, and $\vec{q} = [q_{0,0}, \dots, q_{l,l}]$. We can re-write the inner product $\langle Q, \mathcal{A}_j \rangle$ using the fact that both \mathcal{A}_j and Q are symmetric.

$$(2.24) \quad \langle Q, \mathcal{A}_j \rangle = \vec{q}^T \cdot \vec{a}_j = \vec{a}_j^T \cdot \vec{q}$$

Then, finally, we can re-write the constraint $p(x) = \vec{b}^T Q \vec{b}$, as the system of linear equations that

$$(2.25) \quad \mathcal{A} = \begin{bmatrix} a_0^T \\ \vdots \\ a_l^T \end{bmatrix} q = \begin{bmatrix} c_0 \\ \vdots \\ c_l \end{bmatrix}$$

Thus, the numerical property of the *SDP* problem is completely captured by the *condition number* of A . Therefore, we will write python code to examine different combinations of bases under different degrees and number of variates in the Preliminaries sections. We will list the polynomial bases that we are interested in the following sections, and will briefly touch upon *Semidefinite Programing* before we present our results.

2.3. Polynomial Basis.

2.4. Solving Semidefinite Program. Toy examples maybe

3. NUMERICAL RESULTS

Proposition 3.1.*Proof.*

□

maybe a theorem

Theorem 3.2.

or an example...

Example 3.3.

pictures are always a good idea...

4. MAYBE SOME PROOFS

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.
- [CK07] E. Cheney and D. Kincaid, *Numerical mathematics and computing*, International student edition, Cengage Learning, 2007.
- [Gol96] Gene Golub, *Matrix computations*, Johns Hopkins University Press, Baltimore, 1996.
- [Lau09] M. Laurent, *Sums of squares, moment matrices and optimization over polynomials*, Emerging applications of algebraic geometry, IMA Vol. Math. Appl., vol. 149, Springer, New York, 2009, pp. 157–270.
- [Rec14] M. Recher, *Zur numerischen Auswirkung von Basiswahlen in der polynomiellen Optimierung*, 2014, Master Thesis, Universität zu Köln.

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