Moment-SOS Method to Solve Nonconvex Optimization Problem

Shuvomoy Das Gupta

March 15, 2017

Contents

Introduction

In this note we are going to talk about the moment-SOS method to solve nonconvex optimization problems. The class of nonconvex optimization problems that moment-SOS can solve is rather large - it can solve[1] any polynomial optimization problem. Polynomial optimization problems are those optimization problems where the objective and constraints are polynomial functions. Consider the following optimization problem:

$$f^* = \begin{pmatrix} \text{minimize} & f(x) & \\ \text{subject to} & f_i(x) \geq 0 & i = 1, \ldots, p \\ & h_i(x) = 0 & i = 1, \ldots, q \\ & x \in \mathbf{R}^n, & \end{pmatrix} \tag{1}$$

where $f(x), f_1(x), \ldots, f_p(x), h_1(x), \ldots, h_q(x)$ are polynomials and $x \in \mathbf{R}^n$ is the decision variable. Immediately we can see that this is a rather large class of problems that contains linear programming, mixed integer linear programming [2] and much more. Before proceeding further let's introduce the related concepts.

[1] subject to suitable conditions

[2] Consider a mixed integer linear programming problem:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & 0 \preceq x \preceq M\mathbf{1} \\ & x \in \mathbf{Z}^n \end{array}$$

where $A \in \mathbf{R}^{p \times n}$ is a full row rank matrix, M is a positive integer, and $\mathbf{1}$ is an n-dimensional vector with all components 1. Now the hardness of the problem comes from the last two constraints $0 \leq x_i \leq M$, and x_i integer all $i = 1, \ldots, n$. We can write them as polynomials: $x_i(x_i - 1) \ldots (x_i - M) = 0$ for all $i = 1, \ldots, n$. So, in polynomial format the problem becomes:

$$\begin{array}{lll} \text{minimize} & c^T x & \\ \text{subject to} & Ax = b & \\ & x_i(x_i - 1) \ldots (x_i - M) = 0 & i = 1, \ldots, n \\ & x \in \mathbf{R}^n. & \end{array}$$

Background

Basic Notation. The set of all nonnegative integers is denoted by $\mathbf{N} = \{0, 1, 2, \ldots\}$. By $\mathbf{N}^n$ we mean the set of all n dimensional vectors with the components of the vectors being nonnegative integers. For example,

$$\mathbf{N}^2 = \{(0,0), (0,1), (0,-1), \ldots, (1,2), \ldots\}.$$

The set of all $n$dimensional nonnegative integers with degree bound d is denoted by

$$\mathbf{N}^n_d = \{\alpha \in \mathbf{N}^n \mid \alpha \succeq 0, |\alpha| \leq d\}.$$

The total number of elements in $\mathbf{N}_d^n$ is $\binom{n+d}{d}$. For example,

$$\begin{aligned} \mathbf{N}_2^2 &= \{\alpha \in \mathbf{N}^2 \mid \alpha \succeq 0, |\alpha| \leq 2\} \\ &= \{(0,0),(0,1),(1,0),(1,1),(0,2),(2,0)\}. \end{aligned}$$

Monomial. A monomial in variables $x = (x_1, x_2, \ldots, x_n)$ is a product of the structure $x^\alpha = (x_1, \ldots, x_n)^{(\alpha_1, \ldots, \alpha^n)} = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$, where $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbf{N}^n$. Degree of a monomial x^α is $|\alpha| = \sum_{i=1}^n \alpha_i$.

The truncated zeta vector. The truncated zeta vector of dimension n and degree d, denoted by $[x]_d$ is a vector containing all the monomials in n variables with degree smaller than d. It is expressed as

$$[x]_d = \left(\underbrace{1}_{\text{degree 0 monomial}}, \overbrace{x_1, x_2, \ldots, x_n}^{\text{degree 1 monomials}}, \underbrace{x_1^2, \ldots x_{n-1}x_n, x_n^2}_{\text{degree 2 monomials}}, \ldots, \overbrace{x_1^d, \ldots, x_n^d}^{\text{degree } d \text{ monomials}} \right).$$

Clearly, total number of elements in $[x]_d$ is same as the cardinality of $\mathbf{N}_d^n$, *i.e.*, $\binom{n+d}{d}$.

For example, if we take $n = 2, d = 2$, we have the truncated zeta vector,

$$[x]_2 = \left(1, x_1, x_2, x_1^2, x_1 x_2, x_2^2\right).$$

Polynomial. A (real) polynomial is an expression that is the sum of finite number of terms with each term being a monomial times a real coefficient. A polynomial f can be represented as

$$f(x) = \sum_{\alpha \in S} f_\alpha x^\alpha,$$

where $S \subset \mathbf{N}^n$ and finite, and for all $\alpha \in S$ the corresponding monomial coefficient f_α is a real number. The set of all real polynomials in $x = (x_1, \ldots, x_n)$ with real coefficients is denoted by $\mathbf{R}[x] = \mathbf{R}[x_1, x_2, \ldots, x_n]$. So we say, $f \in \mathbf{R}[x]$. The degree of a polynomial is the maximum degree over all its constituent monomials. Degree of a polynomial f is denoted by degree(f).

For example $f = -3x_1x_2^2x_3 + 4x_3^2 + 5x_1x_3^3$ is a polynomial in $\mathbf{R}[x_1, x_2, x_3]$. It has 3 monomials $x_1x_2^2x_3$ with coefficient -3 and degree 4, x_3^2 with coefficient 5 and degree 2 and $x_1x_3^3$ with coefficient 5 and degree 4. The maximum degree over the monomials is 4, so degree$(f) = 4$.

Sum of squares (SOS) polynomials. A polynomial is a sum of squares if it can be written as a sum of squares of a finite number of polynomials. Consider a polynomial $p \in \mathbf{R}[x]$. It is a sum of squares if there exists a finite number of polynomials $q_1, q_2, \ldots, q_m \in \mathbf{R}[x]$, such that

$$p(x) = \sum_{i=1}^m q_i(x)^2.$$

Naturally p needs to have an even degree and the constituent polynomials would have a degree smaller than or equal to half of degree(p).

For example, $(x_1^3x_2 - 3x_3x_5^2)^2 + (2x_3x_4 + 7x_1x_5 + x_5)^2$ is an SOS polynomial. The set of all SOS polynomials is denoted by $\Sigma[x]$.

Riesz linear functional and moment vector. If $f(x) = \sum_{\alpha \in S} f_\alpha x^\alpha$, then its Riesz linear functional is $L_y(f) = \sum_{\alpha \in S} f_\alpha y_\alpha$ with $y = (y_\alpha)_{\alpha \in S}$ being a new decision vector, which is called moment vector.

For example, consider the polynomial

$$\begin{aligned} f &= -3x_1x_2^2x_3 + 5x_1x_3^3 + 4x_3^2 \\ &= -3(x_1, x_2, x_3)^{(1,2,1)} + 5(x_1, x_2, x_3)^{(1,0,3)} + 4(x_1, x_2, x_3)^{(0,0,2)} \end{aligned}$$

for this the Riesz linear functional is

$$L_y(f) = -3y_{(1,2,1)} + 5y_{(1,0,3)} + 4y_{(0,0,2)},$$

where now we have a linear functional now in the new variables

$$y = \left(y_{(1,2,1)}, y_{(1,0,3)}, y_{(0,0,2)}\right).$$

As there are finite number of elements in y we could keep a table where $(1,2,1), (1,0,3), (0,0,2)$ correspond to $1,2,3$ respectively and write y as $y = (y_1, y_2, y_3)$ a 3 dimensional vector. So by using Riesz linear functional we have expressed a polynomial in terms of a moment vector. This will help us in reformulating a polynomial optimization problem in terms of the moment vector.

Moment matrix. Consider a truncated zeta vector $[x]_d$ with n variables and degree d. Construct the matrix $X_d = [x]_d[x]_d^T$, which by construction is positive semidefinite. Now to each element of of X_d we apply the Riesz linear functional. In the resultant matrix each element is expressed in terms of a moment vector y now (see the example below); it is called a (truncated) moment matrix denoted by $M_d(y)$. So,

$$M_d(y) = L_y(X_d) = L_y\left([x]_d[x]_d^T\right).$$

For example, take $n = 2, d = 2$. As discussed before the truncated zeta vector,

$$[x]_2 = \left(1, x_1, x_2, x_1^2, x_1x_2, x_2^2\right).$$

So,

$$\begin{aligned}
X_2 &= [x]_2[x]_2^T \\
&= \begin{bmatrix} 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1x_2 \\ x_2^2 \end{bmatrix} \\
&= \begin{bmatrix}
1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 \\
x_1 & x_1^2 & x_1x_2 & x_1^3 & x_1^2x_2 & x_1x_2^2 \\
x_2 & x_1x_2 & x_2^2 & x_1^2x_2 & x_1x_2^2 & x_2^3 \\
x_1^2 & x_1^3 & x_1^2x_2 & x_1^4 & x_1^3x_2 & x_1^2x_2^2 \\
x_1x_2 & x_1^2x_2 & x_1x_2^2 & x_1^3x_2 & x_1^2x_2^2 & x_1x_2^3 \\
x_2^2 & x_1x_2^2 & x_2^3 & x_1^2x_2^2 & x_1x_2^3 & x_2^4
\end{bmatrix}.
\end{aligned}$$

Now if we apply the Riesze linear functional element wise to X_2, then we get $M_2(y)$[3]. Similarly, applying L_y to the rest of the elements, we have

$$M_2(y) = \begin{bmatrix}
y_{00} & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\
y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\
y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\
y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\
y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\
y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04}
\end{bmatrix}.$$

[3] For example, consider the (6,5)th and (5,6)th element (both same, as X_2 is symmetric) of X_2, $x_1x_2^3 = (x_1, x_2)^{(1,3)}$, for it $L_y(x_1x_2^3) = y_{(1,3)} = y_{13}$ (to simplify the notation).

Localizing matrix. Consider a polynomial of n variables denoted by $g \in \mathbf{R}[x]$, where $g(x) = \sum_{\gamma \in S} g_\gamma x^\gamma$. Take a truncated zeta vector of n variables (same as g) and degree d. Then the localizing matrix with respect to polynomial g is defined as:

$$M_d\left(gy\right) = L_y\left(g(x)X_d\right) = L_y\left(g(x)[x]_d[x]_d^T\right),$$

where the Riesz linear functional is applied element wise as we did in moment matrix.

For example, take $g(x) = 1 - x_1^2 - x_2^2 \in \mathbf{R}[(x_1, x_2)]$, so $n = 2$. Take the truncated zeta vector $[x]_d = (1, x_1, x_2)$. Now,

$$\begin{aligned}
&g(x)X_2 \\
=&(1 - x_1^2 - x_2^2)\begin{bmatrix} 1 & x_1 & x_2 \end{bmatrix}\begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \\
=&\begin{bmatrix}
-x_1^2 - x_2^2 + 1 & -x_1^3 - x_2^2x_1 + x_1 & -x_2^3 - x_1^2x_2 + x_2 \\
-x_1^3 - x_2^2x_1 + x_1 & -x_1^4 - x_2^2x_1^2 + x_1^2 & -x_2x_1^3 - x_2^3x_1 + x_2x_1 \\
-x_2^3 - x_1^2x_2 + x_2 & -x_2x_1^3 - x_2^3x_1 + x_2x_1 & -x_2^4 - x_1^2x_2^2 + x_2^2
\end{bmatrix}.
\end{aligned}$$

Now we apply the Riesz linear functional $L_y(\cdot)$ to each of the elements of $g(x)X_d$[4] and arrive at the localizing matrix below

$$M_2(gy) = \begin{bmatrix} y_{00} - y_{20} - y_{02} & y_{10} - y_{30} - y_{12} & y_{01} - y_{21} - y_{03} \\ y_{10} - y_{30} - y_{12} & y_{20} - y_{40} - y_{22} & y_{11} - y_{31} - y_{13} \\ y_{01} - y_{21} - y_{03} & y_{11} - y_{31} - y_{13} & y_{02} - y_{22} - y_{04} \end{bmatrix}.$$

[4] For example, the (1,3)th and (3,1) th element (both same, as $g(x)X_2$ is symmetric) of $g(x)X_2$ is

$$\begin{aligned} & -x_2^3 - x_1^2 x_2 + x_2 \\ = & -(x_1, x_2)^{(0,3)} - (x_1, x_2)^{(2,1)} + (x_1, x_2)^{(0,1)}. \end{aligned}$$

So, if we apply $L_y(\cdot)$ on it, we have

$$\begin{aligned} & L_y\left(-(x_1, x_2)^{(0,3)} - (x_1, x_2)^{(2,1)} + (x_1, x_2)^{(0,1)}\right) \\ = & -y_{(0,3)} - y_{(2,1)} + y_{(0,1)} \\ = & -y_{03} - y_{21} + y_{01} \quad \text{(simplifying the notation)} \\ = & y_{01} - y_{21} - y_{03}. \end{aligned}$$

Rewriting the optimization problem. For convenience, we write the original polynomial optimization problem 1 as:

$$f^* = \begin{pmatrix} \text{minimize} & f(x) \\ \text{subject to} & g_j(x) \geq 0 \quad i = 1, \ldots, m \\ & x \in \mathbf{R}^n, \end{pmatrix}, \tag{2}$$

where we have written the equality constraints $h_j(x) = 0$ in problem 1 as two inequality constraints: $h_j(x) \geq 0$ and $h_j(x) \leq 0$, and then renamed all the constraints as $g_j(x)$. Assume the degree of polynomial $g_j(x)$ is either $2v_j$ (even) or $2v_j - 1$ (odd).

Let us simplify even more. Let K be the constraint set, i.e.,

$$K = \{x \in \mathbf{R}^n \mid g_j(x) \geq 0, \quad i = 1, \ldots, m\}. \tag{3}$$

We define an algebraic structure called *quadratic module* on K, denoted by $Q(g_1, \ldots, g_m)$ as follows.

$$Q(g_1, \ldots, g_m) = \left\{\sigma_0 + \sum_{j=1}^{m} \sigma_j g_j \mid \sigma_0, \sigma_1, \ldots, \sigma_m \text{ are sum of squares polynomials}\right\}$$

Then we can write problem 2 as

$$f^* = \begin{pmatrix} \text{minimize} & f(x) \\ \text{subject to} & x \in K \end{pmatrix}. \tag{4}$$

How do we solve this optimization problem? Using *moment-SOS* method. It is due to Lasserre, so it is also called *Lasserre SDP hierarchy*.

When can we apply moment-SOS method? Moment-SOS method applies a famous result from algebraic geometry called *Putinar Positivstellensatz (P-satz)*.

Theorem 1. *(Putinar P-satz) Suppose there exists some polynomial in $Q(g_1, \ldots, g_m)$ such that the super-level set at 0 is compact, i.e., $\{x \in \mathbf{R}^n \mid u(x) \geq 0\}$ is compact*[5]*. If $p(x)$ is some strictly positive polynomial on K, i.e.,*

$$(\forall x \in K) \quad p(x) > 0,$$

then $p \in Q(g_1, \ldots, g_m)$.

[5] If this is so then $Q(g_1, \ldots, g_m)$ is called an Archimedean quadratic module.

To apply moment-SOS method we have to ensure that Putinar P-satz holds. In practice we can make this happen relatively easily. In many real life problems we have some known bound on x for the optimization problem, i.e., we would know that there exists some positive number (possibly large) N such that $\|x\|^2 \leq N$. If we add this constraint to K (because this is a redundant but valid constraint, K does not change), then it makes the quadratic module Archimedean, and Putinar P-satz holds.

The moment-SOS method

Now that we have ensured that Putinar P-satz holds, let us describe the moment-SOS method. In short, the moment-SOS method is a hierarchy of semidefinite optimization problems (hence convex), where the optimal values of the associated optimization problems form monotone nondecreasing sequence of lower bounds that converges to the global optimum of the nonconvex optimization problem. Roughly speaking, moment-SOS method asymptotically approaches the global optimal solution of the nonconvex optimization problem. The algorithm is given by Algorithm 1. For the algorithm we have the following convergence result.

Theorem 2. *(Convergence of Algorithm 1) Suppose Putinar P-satz holds. Then in Algorithm 1*

- $\rho^{(k)} \to f^*$ as $k \to \infty$
- Suppose, the input optimization problem in Algorithm 1 has a global unique minimizer x^*. Then,

$$(\forall j \in \{1, \ldots, n\}) \quad L_{y^{*(k)}}\left(x_j\right) \to x_j^* \text{as } k \to \infty.$$

Algorithm 1 Moment-SOS Algorithm

Input.

• The optimization problem

$$f^* = \begin{pmatrix} \text{minimize} & f(x) & \\ \text{subject to} & g_j(x) \geq 0 & j = 1, \dots, m \\ & x \in \mathbf{R}^n, & \end{pmatrix},$$

where $f(x), g_1(x), \dots, g_m(x)$ are polynomials.

• Maximum number of iterations in the moment-SOS method, k.

Output.

• The optimal value f^*, *or,*

• A nontrivial lower bound $\rho \leq f^*$.

Notation.

•$v_j = \left\lceil \frac{\deg(g_j)}{2} \right\rceil, \quad j = 1, \dots, m$

•$v = \max_{j \in \{1,\dots,m\}} v_j$

•$d_0 = \max \left\{ v, \left\lceil \frac{\deg(f)}{2} \right\rceil \right\}$

Algorithm.

for $d = d_0, d_0 + 1, \dots, k - 1, k$

Solve the d-th semidefinite relaxation problem

$$\rho^{(d)} = \begin{pmatrix} \text{minimize}_y & L_y(f) & \\ \text{subject to} & M_d(y) \succeq 0 & \\ & M_{d-v_j}(g_j y) \succeq 0, & j = 1, \dots, m \\ & y_0 = 1 & \\ & y \in \mathbf{R}^{\binom{n+2d}{2d}} & \end{pmatrix}.$$

Denote an optimal solution to the problem above by $y^{*(d)}$.

if rank $M_{d-v}\left(y^{*(d)}\right) =$ **rank** $M_d\left(y^{*(d)}\right)$

  then $f^* := \rho^{(d)}$

  break

else

  $\rho := \rho^{(d)}$

end if

end for