Determining Non-Negativity of Polynomials using Sum of Squares

Abiyaz Chowdhury, Allen Kim

The main goal of our project is to implement a sum of squares tester for arbitrary polynomials. A polynomial $p(x)$ is said to be nonnegative or positive semidefinite if $p(x) \geq 0$ for all $x \in \mathbb{R}^n$. We also say that a polynomial $p(x)$ is a sum of squares (sos) if it can be represented as a sum of square polynomials, meaning that there exists polynomials $q_1(x), \ldots, q_m(x)$ such that $p(x) = \sum_{i=1}^{m} q_i^2(x)$. It is clear that if a polynomial is sos, then it is nonnegative. However, the converse is not necessarily true in general. Determining whether a polynomial is nonnegative is NP-hard even when the degree of the polynomial is as small as four. However, determining whether a polynomial has a sos decomposition is doable. In general, this acts as a tractable substitute for non-negativity.

An important theorem tells us that a multivariate polynomial $p(x)$ in n variables and degree $2d$ is a sum of squares if and only if there exists a positive semidefinite matrix Q such that

$$p(x) = z^T Q z$$

where z is the vector of monomials of degree up to d

$$z = [1, x_1, x_2, \ldots, x_n, x_1 x_2, \ldots, x_n^d]$$

For our project, we aim to determine whether a given polynomial yields a sum of squares decomposition algorithmically. We outline the steps as follows:

1. Convert the given polynomial as a feasibility problem using semidefinite program.

   We plan to use SymPy to take an arbitrary polynomial as input. Given the coefficients of p, we can expand $z^T Q z$ and match the coefficients to get linear constraints on Q. If we let q be the vectors of Q stacked on top of each other vertically, we can represent these linear constraints as $Aq = b$. Thus, we get the SDP:

   Given a matrix A and vector b, find $Q \geq 0$ such that $Aq = b$.

2. Solve the semidefinite program using interior point methods.

   We hope to implement some interior point method using log barrier functions to solve the SDP.

3. Convert the solution to the SDP back into a sum of squares (if one exists).

   Once we have a Q, since Q is a PSD matrix, we can compute its Cholesky decomposition to get $Q = V^T V$ and then, output the sum of squares directly:

$$p(x) = \sum_i (Vz)_i^2$$