ORFE 523: Convex and Conic Optimization
Final Exam

Zachary Hervieux-Moore

Monday 22^{nd} May, 2017

I pledge my homor that I have not violated the Honor Code or the rules specified by the instructor during this assignment.
Signature: *Zachary Hervieux-Moore*

Exercise 1: Prove or disprove the following statement: Any (multivariate) polynomial can be written as the difference of two sum of squares polynomials.

Answer: First, let us prove this for the one variable case. Consider $c_n x^n$, we can write it as

$$c_n x^n = \left(c_n x^n + \frac{1}{4}\right)^2 - \left(c_n x^n - \frac{1}{4}\right)^2$$

Which is a difference of sum of squares polynomials. Now, for the multivariate case, we can do this for all the terms that only have one variable. Likewise, we can use the same trick to get for the general polynomial

$$c_j x_1^{n_1} x_2^{n_2} \cdots x_n^{n_n} = \left(\underbrace{c_j x_1^{n_1} x_2^{n_2} \cdots x_n^{n_n} + \frac{1}{4}}_{f_j(x)}\right)^2 - \left(\underbrace{c_j x_1^{n_1} x_2^{n_2} \cdots x_n^{n_n} - \frac{1}{4}}_{g_j(x)}\right)^2$$

Where n_i's are the exponents for the i^{th} variable and this is the j^{th} term in the polynomial. Then, we can do this for every term and add them together to get a difference of SOS polynomials

$$P(x) = \sum_j f_j^2(x) - \sum_j g_j^2(x)$$

Exercise 2: A study has provided us with m data points $(x_i, y_i)_{i=1,\ldots,m}$, where x_i is a 2×1 vector and y_i is a scalar. These data points are known to be noisy samples of a bivariate quartic polynomial, which is nonnegative (globally) and convex when restricted to the line passing through the points $(0,1)$ and $(1,1)$. Our goal is to find a bivariate quartic polynomial p which:

i) satisfies the nonnegativity and convexity-along-the-line requirements,

ii) minimizes the least absolute deviations objective function $\sum_{i=1}^{m} |y_i - p(x_i)|$

Write a semidefinite program that finds such a p and carefully argue why your formulation is correct.

Answer: First, let us note that the polynomial we wish to find is in 2 variables and has degree 4. Thus, by a theorem from Hilbert, we have that this polynomial is nonnegative if and only if it can be written as a sum of squares. Now, let us write the optimization problem and slowly change it into an SDP. We have the following optimization

$$\begin{gathered}\min_{p(x)} \sum_{i=1}^{m} |y_i - p(x_i)| \\ \text{s.t. } p(x) \geq 0 \\ \nabla^2 p(x_1, 1) \geq 0\end{gathered}$$

Where the Hessian, $\nabla^2 p(x_1, 1)$, is univariate since the line that goes through the points $(0,1)$ and $(1,1)$ is $x_2 = 1$. This Hessian ensures the convexity along the line constraint as a C^2 function is convex iff its Hessian is PSD. First, we get rid of the absolute value by introducing a new set of variables $\gamma \in \mathbb{R}^m$.

$$\begin{gathered}\min_{\gamma, p(x)} \sum_{i=1}^{m} \gamma_i \\ \text{s.t. } p(x) \geq 0 \\ \nabla^2 p(x_1, 1) \geq 0 \\ -\gamma_i \leq y_i - p(x_i) \leq \gamma_i \\ \gamma_i \geq 0\end{gathered}$$

Where the last two constraints is the standard way of encoding absolute values in the objective. Now, using the result previously mentioned, we make both the polynomials $p(x)$ and $\nabla^2 p(x_1, 1)$ nonnegative by introducing the matrices $Q_1 \in \mathbb{S}^{6\times 6}$ which is associated with the monomial vector $z_1 = (1, x_1, x_2, x_1^2, x_2^2, x_1x_2)^T$ and $Q_2 \in \mathbb{S}^{2\times 2}$ associated with monomial vector $z_2 = (1, x_1)$. Recall, the Hessian $\nabla^2 p(x_1, 1)$ will be a quadratic polynomial in x_1. Furthermore, the terms in Q_2 will be some linear function of the terms in Q_1 depending on your choice of monomial vectors. Thus, changing the nonnegative constraints to SOS constraints we get,

$$\min_{\gamma, Q_1, Q_2} \sum_{i=1}^{m} \gamma_i$$
$$\text{s.t.}$$
$$\text{SOS constraints:}$$
$$Q_1 \succeq 0$$
$$Q_2 \succeq 0$$
$$\text{Linear constraints relating } Q_1 \text{ and } Q_2\text{:}$$
$$Q_{2_{11}} = 4Q_{1_{14}} + 2Q_{1_{22}} + 4Q_{1_{26}} + 4Q_{1_{34}} + 4Q_{1_{45}} + 2Q_{1_{66}}$$
$$Q_{2_{12}} = 6Q_{1_{24}} + 6Q_{1_{46}}$$
$$Q_{2_{22}} = 12Q_{1_{44}}$$
$$\text{Absolute deviation constraints:}$$
$$-\gamma_i \leq y_i - z(x_i)^T Q_1 z(x_i) \leq \gamma_i$$
$$\gamma_i \geq 0$$

This is an SDP as we have PSD constraints coupled with linear constraints and equalities. It is also correct because each transformation we did was iff thanks to Hilbert's theorem. Finally, proof of concept in YALMIP is attached below.

Code Appendix:

```
clear all;
clc;

n = 500;
X = zeros(2,n);
Y = zeros(1,n);

for i=1:n
    X(:,i) = -1 + (2)*rand(2,1);
    error = -0.0001 + 0.0002*rand();
```

```
    Y(i) = 5*X(1,i)^2+0.5*X(1,i)^2*X(2,i)^2+3*X(1,i)^4+2*X(2,i)^2+X(2,i)
        ^4+7+error;
end

sdpvar x1 x2
x = [x1;x2];

sdpvar g(500,1)

d=4;

[p, cp, mp] = polynomial(x,d)

% Polynomial should be positive
F = [sos(p)];

% Convexity along the line x2 = 1
F = F + [sos(hessian(replace(p,x2,1),x1))];

% Coefficients are decision variables
F = F + [cp];

for i=1:n
    F = F + [-g(i) <= Y(i) - replace(p,[x1;x2], X(:,i))];
    F = F + [g(i) >= Y(i) - replace(p,[x1;x2], X(:,i))];
    F = F + [g(i) >= 0];
end

options = sdpsettings('verbose',2,'solver','sdpt3');

[info,z,Q] = solvesos(F,sum(g),options);

sol_Q=Q{1} % get the Gram matrix
sol = (z{1}'*sol_Q*z{1}) % get polynomial
sdisplay(sol)
sdisplay(hessian(replace(sol,x2,1),x1))
```

Exercise 3: For each of the following two matrices

$$A = \begin{bmatrix} 30 & 20 & 17 & 13 \\ 22 & 13 & 0 & -6 \\ 6 & -4 & 39 & 33 \\ 11 & -3 & 28 & 41 \end{bmatrix}, B = \begin{bmatrix} 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \end{bmatrix}$$

prove or disprove the claim that they are copositive.

Answer: First, let us use the fact that if we can decompose $A = C + D$ where C is positive semidefinite and D is copositive, then we have proven that A is copositive as $x^T Cx \geq 0$ and $x^T Dx \geq 0$ for all $x \geq 0$, so $x^T Ax \geq 0$. Thus, we do the decomposition of

$$\begin{bmatrix} 30 & 20 & 17 & 13 \\ 22 & 13 & 0 & -6 \\ 6 & -4 & 39 & 33 \\ 11 & -3 & 28 & 41 \end{bmatrix} = \underbrace{\begin{bmatrix} 30 & 10 & 6 & 11 \\ 10 & 13 & -4 & -6 \\ 6 & -4 & 39 & 28 \\ 11 & -6 & 28 & 41 \end{bmatrix}}_{=C} + \underbrace{\begin{bmatrix} 0 & 10 & 11 & 2 \\ 12 & 0 & 4 & 0 \\ 0 & 0 & 0 & 5 \\ 0 & 3 & 0 & 0 \end{bmatrix}}_{=D}$$

We have that the eigenvalues of C are $(5.108, 12.785, 33.270, 71.837)$ which are all positive and so C is PSD. D is nonnegative and hence trivially copositive. Thus, we conclude that A is copositive.

Now with B, we have

$$x^T Bx = x_1^2 - 2x_1x_2 + 2x_1x_3 + 2x_1x_4 - 2x_1x_5 + x_2^2 - 2x_2x_3 + 2x_2x_4 + 2x_2x_5 \\ +x_3^2 - 2x_3x_4 + 2x_3x_5 + x_4^2 - 2x_4x_5 + x_5^2$$

This has a convenient form as most of the terms can be expressed by a single square. One of the possibilities for the square is

$$\begin{aligned} x^T Bx &= (x_1 - x_2 + x_3 + x_4 - x_5)^2 + 4x_2x_4 - 4x_3x_4 + 4x_3x_5 \\ &= (x_1 - x_2 + x_3 + x_4 - x_5)^2 + 4x_2x_4 + 4x_3(x_5 - x_4) \end{aligned}$$

Thus, if $x \geq 0$, then we have shown that $x^T Bx \geq 0$ if $x_5 \geq x_4$. As all the terms above are positive except for the last when when $x_4 > x_5$. However, we can also use the following factoring

$$\begin{aligned} x^T Bx &= (x_1 - x_2 + x_3 - x_4 + x_5)^2 + 4x_1x_4 - 4x_1x_5 + 4x_2x_5 \\ &= (x_1 - x_2 + x_3 - x_4 + x_5)^2 + 4x_1(x_4 - x_5) + 4x_2x_5 \end{aligned}$$

Thus, if $x \geq 0$, then we have that $x^T Bx \geq 0$ if $x_4 \geq x_5$. Putting the two results together, we have that if $x \geq 0$ then $x^T Bx \geq 0$ as the requirement of $x_4 \geq x_5 \vee x_5 \geq x_4$ is always satisfied.

Exercise 4: A polytope (i.e., a bounded polyhedron) P in $\mathbb{R}^n$ can be represented either through a facet description (as the feasible set of finitely many affine inequalities) or through a vertex description (as the convex hull of a finite set of points in $\mathbb{R}^n$). Given two polytopes P_1 and P_2, we would like to design an algorithm that checks if $P_1 \subseteq P_2$. There are four possibilities here based on the facet/vertex description of each polytope. Out of these four, for how many can you propose an algorithm whose worst-case running time is not exponential in the dimension n, or the number of inputs facets, or the number of input vertices? (Hint: do not be overly greedy, unless you are going for fame and fortune.)

Answer: Let us consider the 4 cases. In all cases, let v_i be the number of vertices of polytope P_i and/or f_i be the number of facets for polytope P_i.

- P_1 vertex, P_2 facet: This has a simple polynomial algorithm. Simply, check if all the vertices in P_1 satisfy all the inequalities in the description of P_2. That is, if $P_2 = \{x \in \mathbb{R}^n : Ax \leq b\}$, simply check that $Ax \leq b$ for all $x \in P_1$. By convexity, if all the vertices (extreme points) of P_1 satisfy the constraint, so does the convex hull, so we have $P_1 \subseteq P_2$. One has to do v_1 matrix multiplication, this has time complexity $O(v_1 n f_1)$ which is polynomial in either input.

- P_1 vertex, P_2 vertex: Similarly to the last one, we want to check if all the v_1 points in P_1 are convex combinations of the points in P_2. Thus, we write a LP that does this for a single point and run it on all the points in P_1's description. Suppose that $x \in P_1$ and $y_i \in P_2$, $i = 1, \ldots, v_2$. This LP is the test to see if x is in the convex hull of the y_i's.

$$\begin{aligned} \min_{\lambda \in \mathbb{R}^{v_2}} \ & \lambda_1 \\ \text{s.t. } \ & 0 \leq \lambda_i \leq 1 \ \forall i = 1, \ldots, v_2 \\ & \sum_{i=1}^{v_2} \lambda_i y_i = x \\ & \sum_{i=1}^{v_2} \lambda_i = 1 \end{aligned}$$

  If this LP returns a valid solution, that means that $\sum_{i=1}^{v_2} \lambda_i y_i = x$ and $\sum_{i=1}^{v_2} \lambda_i = 1$ and so we have that x is in the convex hull of P_2. If all the vertices $x \in P_1$ satisfy this, by convexity, its convex hull is in P_2, or $P_1 \subseteq P_2$. Thus, we need to do this for every point $x \in P_1$ which is v_1 calls. As solving LP's take polynomial time, we have that this algorithm takes polynomial time. Note, the objective is just a dummy variable for illustrative purposes.

- P_1 facet, P_2 facet: In this case, we are given two sets of the forms $P_i = \{x \in \mathbb{R}^n : A_i x \leq b_i\}$. We wish to construct an LP that will test the inclusion. Let A_{2_i} be the i^{th} row of A_2. Then we construct the following LP

$$f_i^* = \max_{x \in \mathbb{R}^n} A_{2_i} x$$
$$A_1 x \leq b_1$$

  If we have that $f_i^* \leq b_{2_i}$ (the i^{th} component of b_2) for all $i = 1, \ldots, f_2$, then we conclude that the polytope described by P_1 is contained by P_2 ($P_1 \subseteq P_2$) since we cannot find a point in P_1 that can violate a single facet of P_2. This algorithm requires f_2 calls of an LP which is polynomial and so we conclude the algorithm is polynomial.

- P_1 facet, P_2 vertex: I will not be overly greedy and so claim that there exists no polynomial time solution for this.

Exercise 5: Consider the following decision problem: Given an $m \times n$ matrix A, an $m \times 1$ vector b, an $n \times n$ symmetric matrix Q, an $n \times 1$ vector d, and a scalar c (all data is assumed to be rational), test whether

$$\{x \in \mathbb{R}^n : Ax \leq b\} \subseteq \{x \in \mathbb{R}^n : x^T Q x + d^T x + c \leq 1\}$$

Show that this problem is NP-hard.

Answer: We will do a reduction from testing copositivity which is known to be NP-hard from class. We let $A = -I$, $b = 0$, $Q = -C$, $d = 0$, and $c = 1$. Note that we can assume C is symmetric wlog because, if it were not, we could add to it a nonnegative matrix which is copositive to make it symmetric. The above problem is then

$$\{x \in \mathbb{R}^n : x \geq 0\} \subseteq \{x \in \mathbb{R}^n : x^T C x \geq 0\}$$

Thus, if the above is true, then C is copositive as $x^T C x \geq 0$ for all $x \geq 0$. Likewise, if C is copositive, then we have the above inclusion is true. That is

$$\{x \in \mathbb{R}^n : x \geq 0\} \subseteq \{x \in \mathbb{R}^n : x^T C x \geq 0\} \iff x^T C x \geq 0 \text{ for all } x \geq 0$$

Thus, since testing copositivity of a matrix C is NP-hard, we have shown that testing this inclusion is also NP-hard. That is, testing copositivity is a special case of testing containment among linear/quadratic basic semialgebraic sets.