CONVEX AND CONIC OPTIMIZATION
Homework 6

NILAVA METYA
nilava.metya@rutgers.edu
nm8188@princeton.edu

May 2, 2024

Problem 1

1. Suppose you had a blackbox that given a 3SAT instance would tell you whether it is satisfiable or not. How can you make polynomially many calls to this blackbox to find a satisfying assignment to any satisfiable instance of 3SAT?
2. Suppose you had a blackbox that given a graph G and an integer k would tell you whether G has a stable set of size larger or equal to k. How can you make polynomially many calls to this blackbox to find a maximum stable set of a given graph?

Solution

1. Denote the blackbox by f. So f takes in a formula (in three variables) and outputs 1 if satisfiable, and 0 otherwise.
   Let S be a formula in CNF, with variables $x_1, \cdots, x_n$. Treat S as a polynomial in x_i's. Assume S is satisfiable, i.e., there are values $a_1, \cdots, a_n \in \{0,1\}$ such that $S(a_1, \cdots, a_n) = 1$. Often we denote $\boldsymbol{a} = (a_1, \cdots, a_n)$. We will make n calls to the blackbox.

   Claim 1
   For each $i \in [n]$, at least one of $f(x_i \wedge S)$ or $f(\overline{x_i} \wedge S)$ is 1.

   Proof. If $a_i = 1$ then $(x_i \wedge S)(\boldsymbol{a}) = 1 \cdot S(\boldsymbol{a}) = 1$. If $a_i = 0$ then $(\overline{x_i} \wedge S)(\boldsymbol{a}) = 1 \cdot S(\boldsymbol{a}) = 1$. ■

   We find each $f(x_i \wedge S)$ with the blackbox. For each i, if $f(x_i \wedge S) = 1$ then set $a_i = 1$, otherwise set $a_i = 0$.
   Denote $y_i \coloneqq \begin{cases} x_i & \text{if } a_i = 1 \\ \overline{x_i} & \text{otherwise} \end{cases}$. $\boldsymbol{a}$ satisfies every $y_i S$ by the above. We'll show that $\boldsymbol{a}$ satisfies S.

$$\begin{aligned}
\left(\bigwedge_{i=1}^{n} (y_i S)\right)(\boldsymbol{a}) &= \left[\left(\bigwedge_{i=1}^{n} y_i\right) \wedge S\right](\boldsymbol{a}) && [\because \alpha \wedge \alpha = \alpha] \\
\implies \prod_{i=1}^{n} (y_i \wedge S)(\boldsymbol{a}) &= \left(\prod_{i=1}^{n} y_i(\boldsymbol{a})\right) \cdot S(\boldsymbol{a}) && [\because (\alpha \wedge \beta)(\boldsymbol{a}) = \alpha(\boldsymbol{a}) \cdot \beta(\boldsymbol{a})] \\
\implies 1 &= \left(\prod_{i=1}^{n} y_i(\boldsymbol{a})\right) \cdot S(\boldsymbol{a}) && [\because (y_i S)(a) = 1 \forall i] \\
\implies 1 &= S(\boldsymbol{a}) && [\because y_i(\boldsymbol{a}) = 1 \forall i \text{ by construction}]
\end{aligned}$$

2.

Problem 2

Consider a family of decision problems indexed by a positive integer k:

RANK-k-SDP

Input: Symmetric $N \times N$ matrices $A_1, \cdots, A_m$ with entries in $\mathbb{Q}$, scalars $b_1, \cdots, b_m \in \mathbb{Q}$.
Question: Is there a real symmetric matrix X that satisfies the constraints

$$\mathrm{Tr}(A_i X) = b_i, i \in [m], X \succeq 0, \mathrm{rank}(X) = k? \tag{1}$$

Solution

RANK-1-SDP:

First notice that a symmetric psd matrix $X \in S^n$ has rank 1 iff it has the form $xx^\top$ for some $x \in \mathbb{R}^n \setminus \{0\}$. So in the underlying problem, the constraints can be rewritten as $b_i = \mathrm{Tr}(A_i X) = x^\top A_i x \forall i \in [m]$ and $x \neq 0$. Thus it is clear that

$$\exists X \in S^n \text{ s.t. } X \succeq 0, \mathrm{Tr}(A_i X) = b_i \forall i \in [m], \mathrm{rank}(X) = 1 \iff \exists x \in \mathbb{R}^n \text{ s.t. } x^\top A_i x = b_i \forall i \in [m], x \neq 0. \tag{2}$$

We will show a reduction **STABLE-SET** $\longrightarrow$ **RANK-1-SDP**. Let (G, k) be an instance of **STABLE-SET** where graph G has edges E and vertices $[n]$. Recall that this is same as feasibility of some $v \in \mathbb{R}^n$ satisfying $v_i(1 - v_i) = 0 \ \forall i \in [n], v_i v_j = 0 \ \forall \{i, j\} \in E, \sum\limits_{i \in [n]} v_i \geq k.$

Since each $v_i \in \{0, 1\}, \sum_{i \in [n]} v_i$ whence the last constraint is equivalent to the existence of some $s \in \mathbb{R}$ such that $\left(\sum\limits_{i \in [n]} v_i\right)^2 - k^2 = s^2$. So **STABLE-SET** is the feasibility of some v, s subject to

$$\begin{aligned} v \in \mathbb{R}^n&, \, s \in \mathbb{R} \\ v_i(1 - v_i) &= 0 \ \forall i \in [n] \\ v_i v_j &= 0 \ \forall \{i, j\} \in E \\ \left(\sum_{i \in [n]} v_i\right)^2 &= k^2 + s^2. \end{aligned} \tag{(Q)}$$

Claim 2
Feasibility of (Q) is equivalent to the feasibility of v, c, s subject to

$$\begin{aligned} v \in \mathbb{R}^n, c \in \mathbb{R}&, \, s \in \mathbb{R} \\ c^2 &= 1 \\ v_i(c - v_i) &= 0 \ \forall i \in [n] \\ v_i v_j &= 0 \ \forall \{i, j\} \in E \\ \left(\sum_{i \in [n]} v_i\right)^2 - s^2 &= k^2. \end{aligned} \tag{(QQ)}$$

Proof. If (v, s). is feasible to (Q), then $(v, 1, s)$ is clearly feasible to (QQ).
Now suppose (v, c, s) is feasible to (QQ). So each $v_i \in \{0, c\} \forall i \in [n]$. This means $k^2 + s^2 = \left(\sum i v_i\right)^2 = \left(\sum_{\substack{i \in [n] \\ v_i = c}} v_i\right)^2 = c^2 \left(\sum_{\substack{i \in [n] \\ v_i \neq 0}} 1\right)^2 = \left(\sum_{\substack{i \in [n] \\ v_i \neq 0}} c v_i\right)^2 = \left(\sum_i c v_i\right)^2$. This shows that (cv, s) is feasible to (QQ). ∎

Notice how all constraints in (QQ) are polynomial expressions with only constant terms and homogeneous quadratic terms. This perfectly matches with what we want in the original problem using eq. (2). To get an instance of **RANK-**1**-SDP**, take:

- The size of matrices $N = n + 2$,
- $Q = e_{n+1}e_{n+1}^\top$ and $q = 1$,
- $B_i = E_{i,n+1} - 2e_ie_i^\top$ and $r_i = 0$ for each $i \in [n]$
- $A_{ij} = E_{ij} = e_ie_j^\top + e_je_i^\top$ (which is the $N \times N$ matrix with 1 in positions $(i,j),(j,i)$ and 0 elsewhere), $b_{ij} = b_{ji} = 0$ for each edge $\{i,j\} \in E$,
- $T = \sum\limits_{1\le i,j\le n} e_ie_j^\top - e_Ne_N^\top, b = k^2$.

The above are clearly all rational.

Now consider the constraints on x

$$\begin{aligned} x &\in \mathbb{R}^N \\ x^\top Qx &= q \\ x^\top B_ix &= r_i \forall i \in [n] \\ x^\top A_{ij}x &= b_{ij} \forall \{i,j\} \in E \\ x^\top Tx &= b \end{aligned} \tag{(HQ)}$$

We show that this answers the same question as **STABLE-SET** gives on (G,k).

Claim 3
(QQ) is feasible iff (HQ) is feasible.

Proof. A solution $(v,c,s) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}$ of (QQ) corresponds to a solution $x = (v,c,s) \in \mathbb{R}^N$ of (HQ).

$$\begin{aligned} x^\top Qx &= c^2 & 1 &= q \\ x^\top B_ix &= 2v_ic - 2v_i^2 & 0 &= r_i \\ x^\top A_{ij}x &= 2v_iv_j & 0 &= b_{ij} \\ x^\top Tx &= (v_1 + \cdots + v_n)^2 - s^2 & k^2 &= b. \end{aligned}$$

Left column matches the LHS and right column matches the RHS of each constraint in (QQ) and (HQ). ■

Notice that every solution x of (HQ) is automatically nonzero because of the first constraint: $x_{n+1}^2 = 1$.

The above shows that G has a stable set of size (at least) k iff (HQ) is feasible, which is a problem of **RANK-**1**-SDP**. This completes the proof that **RANK-**1**-SDP** is NP hard because we showed a reduction from an NP hard problem.

RANK-k-SDP:

We will show a reduction **RANK-**1**-SDP** $\longrightarrow$ **RANK-k-SDP** for $k \geq 2$. Indeed, consider an instance of **RANK-**1**-SDP** with inputs which are $N \times N$ matrices $A_1, \cdots, A_m \in \mathbb{Q}^{N\times N}$ and scalars $b_1, \cdots, b_m \in \mathbb{Q}$. Consider the following constraints:

$$
\begin{aligned}
X \in & S^{N+k-1} \\
\operatorname{rank}(X) &= k \\
X &\succeq 0 \\
\operatorname{Tr}(E_{ij}X) &= 0 \ \forall N < i < N+k, 1 \leq j \leq N \\
\operatorname{Tr}(E_{ij}X) &= 0 \ \forall N < i < j < N+k \\
\operatorname{Tr}(E_{ii}X) &= 0 \ \forall N < i < N+k \\
\operatorname{Tr}(\tilde{A}_i X) &= b_i \ \forall i \in [m].
\end{aligned}
\tag{3}
$$

Here $\tilde{A}_i = \begin{bmatrix} A_i & 0 \\ 0 & 0 \end{bmatrix} \in S^{N+k-1}$ obtained by putting appropriate number of 0's.

We discuss the above linear constraints one by one. On slight inspection, the above linear constraints deal with four different blocks of $X = \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix}$ where $A \in S^N, B \in \mathbb{R}^{N \times (k-1)}, C \in S^{k-1}$.

- $\operatorname{Tr}(E_{ij}X) = 0 \ \forall N < i < N+k, 1 \leq j \leq N$: here $E_{ij} = e_i e_j^\top + e_j e_i^\top$. This is just saying that $X_{ij} = X_{ji} = 0$ whenever $i > N$ and $j \in [N]$. In other words $B = 0$.
- $\operatorname{Tr}(E_{ij}X) = 0 \ \forall N < i < j < N+k$: This looks at the block C because $j > i > N$. The constraint says that all off-diagonal entries of C are 0, that is, C is a diagonal matrix.
- $\operatorname{Tr}(E_{ii}X) = 0 \ \forall N < i < N+k$: This says that the diagonal entries of C are all 1. Using the previous point, C is the identity matrix of size $(k-1) \times (k-1)$.

So this makes X look like $\begin{bmatrix} A & 0 \\ 0 & I_{k-1} \end{bmatrix}$. Note that $\operatorname{rank}(X) = \operatorname{rank}(A) + \operatorname{rank}(I_{k-1}) = \operatorname{rank}(A) + k - 1$, whence $\operatorname{rank}(X) = k \iff \operatorname{rank}(A) = 1$. Further $\begin{bmatrix} A & 0 \\ 0 & I_{k-1} \end{bmatrix} \succeq 0 \iff A \succeq 0$ because eigenvalues of a block matrix formed of two blocks stacked diagonally is the union of the eigenvalues of the two individual blocks.

Claim 4
1 is feasible iff 3 is feasible.

Proof. From the above bullet points and the rank discussion, it is clear that a rank 1 solution $A \succeq 0$ of 1 corresponds to a rank k solution $\begin{bmatrix} A & 0 \\ 0 & I_{k-1} \end{bmatrix} \succeq 0$. ∎

This completes the proof that **RANK-k-SDP** is NP hard because we showed a reduction from an NP hard problem.

Problem 3

A polynomial $p(x) := p(x_1, \cdots, x_n)$ is nondecreasing with respect to a variable x_i if $\frac{\partial p}{\partial x_i}(x) \geq 0 \forall x \in \mathbb{R}^n$. Show that the problem of deciding whether a degree-d polynomial with rational coefficients is nondecreasing with respect to a particular variable (e.g., x_1) is

(i) in P if $d < 5$.

(ii) NP-hard if $d \geq 5$.

Solution

Set $d = \deg(p)$. Here $p \in \mathbb{Q}[x_1, \cdots, x_n]$. For short, denote $\partial_i p := \frac{\partial p}{\partial x_i}$.

First note that $\deg(\partial_i(p)(x)) < d$ for each $i \in [n]$. This is because (fixing some $i \in [n]$) if $p(x) = x_i^k q_k(x_{-i}) + \cdots + x_i q_1(x_{-i}) + q_0(x_{-i})$ with all $q_j \in \mathbb{Q}[x_{-i}]$ (x_{-i} denotes the vector with all variables in x without x_i), then $d = \max\limits_{0 \leq j \leq k} \{j + \deg(q_j)\}$ whence $\deg(\partial_i p) = \max\limits_{1 \leq j \leq k} \{j - 1 + \deg(q_j)\} \leq \max\limits_{0 \leq j \leq k} \{j - 1 + \deg(q_j)\} = d - 1$ as a polynomial in $\mathbb{Q}[x]$.

The problem as mentioned, for each degree d, takes input n, the number of variables, the rational coefficients that make a polynomial, and an index i; and answers the question if the polynomial is non-decreasing in variable x_i. Denote the input size by N - that is the number of coefficients till degree d. The degree d of this polynomial can in fact be found in polynomial time in N. So there is a problem for each degree. Call it **MONO-d**.

(i) We will show this by cases on degree of $d' = \deg(\partial_i p(x))$. Note that $d' \leq 4$ if $d \leq 5$.

$d = 0$: Then $\partial_i p = 0 \geq 0$. So non-increasing. Thus we answered in constant time.

$d' = 0$: Then $\exists b \in \mathbb{Q}$ such that $\partial_i p(x) = b$. Non-negativity of the rational constant b can be checked in constant time.

$d' = 1$: The required derivative looks like $\partial_i p(x) = b^\top x + c$ for some $b \in \mathbb{Q}^n, c \in \mathbb{Q}$. This is ≥ 0 everywhere on $\mathbb{R}^n$ iff $b \neq 0$ and $c \geq 0$. Nonzero-ness of b can be checked in linear (in n) time and $c \geq 0$ can be checked in constant time.

$d' = 2$: The required derivative looks like $\partial_i p(x) = \frac{1}{2} x^\top A x - b^\top x + c$ for some $A \in \mathbb{Q}^{n \times n}, b \in \mathbb{Q}^n, c \in \mathbb{Q}$ and A symmetric. Then:

- Say $A \not\succeq 0$. This is checked in $\mathcal{O}(n^3)$ time. Then the function $\partial_i p$ is unbounded below: along the direction of some eigenvector with negative eigenvalue.
- So $A \succeq 0$ now. We have reduced to the problem of minimizing the convex function $\partial_i p(x)$ where $x \in \mathbb{R}^n$. Recall that if $x \in \mathbb{R}^n$ is a minima of this function then it's a critical point. Conversely if $x \in \mathbb{R}^n$ is a critical point, then $\nabla(\partial_i p)(x) = 0 \implies \nabla(\partial_i p(x))(y - x) \geq 0 \forall y \in \mathbb{R}^n$ whence it is a minima (convexity was needed here). If there is no $x \in \mathbb{R}^n$ such that $(Ax - b =)\partial_i p(x) = 0$, then there is no minima and the problem is unbounded below. Otherwise assume there is a critical point $v \in \mathbb{R}^n$, that is, $Av = b$. In other words, b is in the column span (= row transpose span because A is symmetric) of A. Then we claim every critical point takes the same objective. Say u is another critical point. Clearly $Av = Au = b$ whence $A^\top(u - v) = A(u - v) = 0$. Since b is in the column span of A, $b^\top(u - v) = 0$. But $\partial_i p(v) = -\frac{1}{2} b^\top v + c = \frac{1}{2} b^\top u + c = p(u)$. It follows that any critical point v gives a global minima with value $-\frac{1}{2} b^\top v + c$.

Thus our polynomial time algorithm is:

(a) Differentiate $p(x)$ to get A, b, c such that $\partial_i p(x) = \frac{1}{2} x^\top A x - b^\top x + c$.

(b) Check if $A \succeq 0$. If not, conclude that p is not non-decreasing wrt x_i and `EXIT` (because $\partial_i p$ is then unbounded below). Otherwise go to the next step.

(c) Check if b is in the column span of A. If not, conclude that p is not non-decreasing wrt x_i and `EXIT` (because $\partial_i p$ is then unbounded below). Otherwise go to the next step.

(d) Otherwise $\{x \in \mathbb{R}^n \mid Ax = b\} \neq \varnothing$. Find a solution $\overline{x}$ to $Ax = b$. The objective $\partial_i p(\overline{x})$ becomes $-\frac{1}{2} b^\top \overline{x} + c$. If this value is ≥ 0, conclude that p is non-decreasing wrt x_i and `EXIT`. Otherwise conclude that p is not non-decreasing wrt x_i and `EXIT`.

This is correct because of the above discussion. This runs in polynomial time in size of input because step (a) takes same number of steps as number of coefficients; step (b) takes $\mathcal{O}(n^3) \leq \mathcal{O}(N^3)$ steps; step (c) can be done in polynomial time (in n) again (say Gaussian elimination); and step (d) can again be done in polynomial steps in n. So overall, the number of steps this algorithm takes is polynomial in the input size N.

$d' = 3:$ So $\partial_i p(x)$ restricted to $\{x \in \mathbb{R}^n \mid x_1 = x_2 = \cdots = x_n\}$ gives a $3-$degree polynomial in one variable which, we know from elementary theory, is unbounded below.

(ii) We'll now show a reduction **COPOS** $\longrightarrow$ **MONO-**d for $d \geq 5$, where **COPOS** takes input a matrix M and decides whether it is copositive. Given an input $M \in \mathbb{Q}^{n \times n}$ to **COPOS**, we'll give the input $p(x_1, \cdots, x_{n+1}) \coloneqq x_{n+1} \left(\sum_{1 \leq i \leq j \leq n} M_{ij} x_i^2 x_j^2 \right) + x_1^d$ and variable index $n+1$ to **MONO-**d. This makes $\deg$ of the first term to be 5 which is why we want $d \geq 5$. Then the following is immediate.

Claim 5
M is copositive iff q is non-decreasing wrt x_{n+1}.

Proof. Note that $\partial_{n+1} p(x) = \sum_{1 \leq i \leq j \leq n} M_{ij} x_i^2 x_j^2 = \begin{bmatrix} x_1^2 \\ \vdots \\ x_n^2 \end{bmatrix}^\top M \begin{bmatrix} x_1^2 \\ \vdots \\ x_n^2 \end{bmatrix} \geq 0$ for all $x \in \mathbb{R}^{n+1}$ iff $v^\top M v \geq 0 \forall v \in \mathbb{R}^n$ such that $v \geq 0$ iff M is copositive. ∎

Since **COPOS** is known to be NP-hard, the above reduction shows that **MONO-**d is NP-hard.

Problem 4

1. In the file `regression_data.mat`, you are given 20 points (x_i, f_i) in $\mathbb{R}^2$ where $(x_i)_{i=1,\cdots,20}$ are the entries of the vector `xvec` and $(f_i)_{i=1,\cdots,20}$ are the entries of the vector `fvec`. The goal is to fit a polynomial of degree 7

$$p(x) = c_0 + c_1 x + \cdots + c_7 x^7 \tag{4}$$

to the data to minimize least square error:

$$\min_{c_1,\cdots,c_7} \sum_{i=1}^{20} (f_i - p(x_i))^2. \tag{5}$$

The data comes from noisy measurements of an unknown function that is a priori known to be nondecreasing (e.g., the number of calories you intake as a function of the number of Big Macs you eat).

(a) If the underlying function is truly monotone and the noise is not too large, one may hope that least squares would automatically respect the monotonicity constraint. Solve (5) to see if this is the case. Plot the optimal polynomial you get and report the optimal value.

(b) Resolve (5) subject to the constraint that the polynomial (4) is nondecreasing. Plot the optimal polynomial you get and report the optimal value.