CONVEX AND CONIC OPTIMIZATION
Endterm

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

May 12, 2024

Problem 1

Suppose a set $S \subseteq \mathbb{R}^n$ is nonempty, convex, and closed and that its complement $\overline{S}$, i.e., the set $\mathbb{R}^n \setminus S$, is convex and nonempty. Show that S must be a halfspace.

Solution

Since $S, \overline{S}$ are both convex, $\exists a \in \mathbb{R}^n \setminus \{0\}, b \in \mathbb{R}$ such that $a^\top x \leq b \; \forall x \in S$ and $a^\top x \geq b \; \forall x \in \overline{S}$ (we could use separating hyperplane theorem because $S, \overline{S}$ are non-empty convex sets that don't intersect). Let $T = \left\{x \in \mathbb{R}^n \;\middle|\; a^\top x \leq b\right\}$. The above shows that $S \subseteq T$. We want to show $T \subseteq S$, in other words, $\overline{S} \subseteq \overline{T} = \left\{x \in \mathbb{R}^n \;\middle|\; a^\top x > b\right\}$.

We are done if we show that $x \in \overline{S} \implies a^\top x \neq b$. For the sake of contradiction, suppose $x \in \overline{S}$ is such that $a^\top x = b$. $\overline{S}$ is open (because S is closed), so $\exists r > 0$ such that $B^o_{2r}(x) \coloneqq \{y \in \mathbb{R}^n \mid \|y - x\|_2 < 2r\} \subseteq \overline{S}$. Consider the vector $y \coloneqq x - \frac{r}{\|a\|} a$. Then $\|y - x\| = r$ whence $y \in B^o_{2r}(x) \subseteq \overline{S}$. However $a^\top y = a^\top \left(x - \frac{r}{\|a\|} a\right) = b - r\|a\| < b$. But we saw earlier that inner product of a with any point in $\overline{S}$ must be at least b. This is a contradiction. It follows that $a^\top x < b \; \forall x \in \overline{S}$. And thus $S = T = \left\{x \in \mathbb{R}^n \;\middle|\; a^\top x \leq b\right\}$ which is a halfspace.

Problem 2

For a graph G on n vertices, let $\alpha(G)$ be the stability number of G, $\vartheta(G)$ be the Lovász theta number of G, and $\vartheta'(G)$ be the optimal value of the semidefinite program:

$$\vartheta'(G) = \begin{cases} \min\limits_{\substack{P \in S^n \\ k \in \mathbb{R}}} \; k \\ \text{s.t. } k(I+A) - J - P \geq 0 \\ \qquad P \succeq 0 \end{cases} \tag{(Q)}$$

(a) Show that $\alpha(G) \leq \vartheta'(G) \leq \vartheta(G)$

(b) Let G be the graph with vertex set corresponding to all the vectors in $\{0,1\}^6$, two distinct vectors being adjacent if their ℓ_1 distance (i.e., Hamming distance) is less than 4. For your convenience, the adjacency matrix of this graph is given in the file `AdjacencyMatrix.mat`. Compute $\alpha(G), \vartheta'(G)$, and $\vartheta(G)$.

Solution

(a) Proof of $\alpha(G) \leq \vartheta'(G)$: Say (P,k) is feasible to (Q). Consider an arbitrary $x \in \mathbb{R}^n$ and $x \geq 0$. Then $kx^\top(I+A)x - x^\top Jx \geq x^\top Px$ because of the first constraint. Using the positive semi-definiteness of P we get $kx^\top(I+A)x \geq x^\top Jx$. However $x^\top Jx = \left(\sum_{i=1}^n x_i\right)^2$. Furthermore, if $x \in \Delta_{n-1} = \{p \in \mathbb{R}^n : p \geq 0, \|p\|_1 = 1\}$, then the above satisfy that $x^\top(I+A)x \geq \frac{1}{k}$.
Recall that the Motzkin-Strauss theorem says that $\dfrac{1}{\alpha(G)} = \min\limits_{x \in \Delta_{n-1}} x^\top(A+I)x$. But we have $x^\top(I+A)x \geq \frac{1}{k}\ \forall x \in \Delta_{n-1}$. Thus $\alpha(G) \leq k$. Here k is itself the objective value of any feasible point in the optimization problem (Q). Since $\alpha(G)$ is a lower bound of the objective of any feasible point in (Q), it must also be a lower bound of the optimal value $\vartheta'(G)$.

Proof of $\vartheta'(G) \leq \vartheta(G)$: **Discussion**: The inequations in (Q) hint at the dual of the Lovasz SDP: $\min\{t : (t,Z) \in \mathbb{R} \times S^n, Z_{ij} = 0 \forall \{i,j\} \notin E, tI + Z \succeq J\}$. This is from equation (7) in Lecture 11. The expectation is that after transforming the PSD constraints in (Q), it will look similar to the PSD constraints of the Lovasz SDP, and the other constraints in (Q) will be a relaxation. The first instinct was to take $kI + Z - J = P$ so that $k(I+A) - J - P = kA - Z$. As an attempt, I also tried taking the dual of this program and it seemed to be close to the original Lovasz SDP but I did not know how to prove that there is no duality gap. So it made sense to start with the dual of the SDP.

Consider (Q) along with the following three optimization problems

$$\begin{aligned} &\min_{\substack{X \in S^n \\ k \in \mathbb{R}}} \; k \\ &\text{s.t. } kA \geq X \\ &\qquad kI + X \succeq J \end{aligned} \tag{(R)}$$

$$\begin{aligned} &\min_{\substack{X \in S^n \\ k \in \mathbb{R}}} \; k \\ &\text{s.t. } X_{ij} \leq 0 \; \forall \{i,j\} \in \overline{E} \\ &\qquad X_{ii} \leq 0 \; \forall i \in [n] \\ &\qquad kI + X \succeq J \end{aligned} \tag{(S)}$$

$$\begin{aligned} &\min_{\substack{Z \in S^n \\ k \in \mathbb{R}}} \; k \\ &\text{s.t. } Z_{ij} = 0 \; \forall \{i,j\} \in \overline{E} \\ &\qquad Z_{ii} = 0 \; \forall i \in [n] \\ &\qquad kI + Z \succeq J \end{aligned} \tag{(L)}$$

Claim 1
optval(Q) = optval(R).

Proof. Take $X = P + J - kI$ to get (R) from (Q) (and take $P = kI + X - J$ for the other way). $k(I + A) - J - P \geq 0 \iff kA \geq X$. $P \succeq 0 \iff kI + X \succeq J$. Objectives are same. ∎

Claim 2
optval$(R) \leq$ optval(S).

Proof. (These values are in fact equal, but we'll only prove the given inequality.)
Say (X, k) is feasible to (S). So $X_{ij} \leq 0 \; \forall \{i, j\} \notin E$ ($\{i, j\} \notin E$ is satisfied for the case $i = j$) and $kI + X - J \succeq 0$. Let's show that (X, k) is feasible to (R). The PSD criteria are the same, so we need not worry about that.
Pick an edge $\{i, j\} \in E$. Then

$$
\begin{aligned}
0 &\leq (e_i - e_j)^\top (kI + X - J)(e_i - e_j) \\
&= (kI_{ii} + X_{ii} - J_{ii}) + (kI_{jj} + X_{jj} - J_{jj}) - 2(kI_{ij} + X_{ij} - J_{ij}) \\
&= (k + X_{ii} - 1) + (k + X_{jj} - 1) - 2(X_{ij} - 1) \\
\implies 2(X_{ij} - 1) &\leq 2(k - 1) + X_{ii} + X_{jj} \leq 2(k - 1) \\
\implies X_{ij} &\leq k
\end{aligned}
$$

Thus if $\{i, j\} \in E$ then $X_{ij} \leq k = k \cdot A_{ij}$ as shown above.[1] If $\{i, j\} \notin E$ then $X_{ij} \leq 0 = k \cdot A_{ij}$ anyway. So $X \leq kA$. ∎

Claim 3
optval$(S) \leq$ optval(L).

Proof. The constraints are simply relaxed from "$= 0$" in (L) to "≤ 0" in (S). So the feasible set of (S) contains that of (L). Minimizing over a larger set (i.e., in (S)) gives (non-strict) smaller values. So the given claim holds. ∎

Combining the above claims gives $\vartheta'(G) =$ optval$(Q) \leq$ optval$(L) = \vartheta(G)$.

(b) See code next page.

The Lovasz SDP gives $\vartheta(G) = 5.\overline{3}$ (rounding to the nearest sensible/intuitive number).

In order to find $\alpha(G)$, we start by checking whether there is a stable set of size 5 (as indicated by Lovasz SDP). That fails. So we check for a stable set of size 4 and we find many. One of them is given by vertices $0, 15, 51, 60$ (indexing at 0). This has been checked in the code: the submatrix corresponding to these rows and columns is all 0. So $\alpha(G) = 4$ because there is no stable set of size 5, but there are stable sets of size 4.

But the given optimization problem has optimal value $\vartheta'(G) = 4$ (since it's $3.99999\cdots$ and we proved that it's an upper bound on $\alpha(G)$ and taking the closest sensible answer keeping in mind numerical errors).

Thus the given optimization solution is tight, while the Lovasz SDP has a gap.

[1] Another direct way to argue is that the maximum absolute value of a PSD matrix occurs on diagonal and the diagonal has non-negative entries so $k - 1 \geq k + X_{ii} - 1 = |k + X_{ii} - 1| \geq |0 + X_{ij} - 1| \geq X_{ij} - 1$.

[1]:
```python
import numpy as np
import cvxpy as cp
import scipy
mat = scipy.io.loadmat('AdjacencyMatrix.mat')
G = mat['A']
```

[2]:
```python
n = len(G)
one = [1 for i in range(n)]
J = np.outer(one, one)
I = np.identity(n)

#auxiliary method to check if the list of vertices v forms a stable set in graph
def isStable(v):
    l = len(v)
    for i in range(l):
        for j in range(i+1,l):
            if(G[v[i]][v[j]] == 1):
                return False
    return True
```

Calculation of $\vartheta(G)$

[3]:
```python
X = cp.Variable((n,n), symmetric = True)
constraints = [X >> 0, cp.trace(X) == 1]
for i in range(n):
    for j in range(i,n):
        if(G[i][j] == 1):
            constraints.append(X[i][j] == 0)
constraints
prob = cp.Problem(cp.Maximize(cp.trace(J @ X)), constraints)
print(prob.solve(),"\n")
```

```
5.333333264771721
```

Calculation of $\alpha(G)$

[4]:
```python
#stable sets of length 5
s5 = 0
for i in range(n):
    for j in range(i+1,n):
        if(G[i,j]==1):
            continue
        for k in range(j+1,n):
            if(G[j,k] == 1 or G[i,k] == 1):
                continue
```

```
            for l in range(k+1,n):
                if(not isStable([i,j,k,l])):
                    continue
                for t in range(l+1,n):
                    if(isStable([i,j,k,l,t])):
                        print([i,j,k,l,t])
                        s5 = s5 + 1
print(s5)
```

0

[5]:
```
#stable sets of length 4
s4 = 0
for i in range(n):
    for j in range(i+1,n):
        if(G[i,j]==1):
            continue
        for k in range(j+1,n):
            if(G[j,k] == 1 or G[i,k] == 1):
                continue
            for l in range(k+1,n):
                if(isStable([i,j,k,l])):
                    #print([i,j,k,l])
                    s4 = s4 + 1
print(s4)
```

240

[6]:
```
stb = [0, 15, 51, 60]
print(np.array([[G[i][j] for i in stb] for j in stb]))
```

```
[[0 0 0 0]
 [0 0 0 0]
 [0 0 0 0]
 [0 0 0 0]]
```

Calculation of $\vartheta'(G)$

Here we solve the following problem:

$$\vartheta'(G) = \begin{cases} \min\limits_{\substack{P \in S^n \\ k \in \mathbb{R}}} \; k \\ \quad \text{s.t. } k(I + A) - J - P \geq 0 \\ \quad\quad\; P \succeq 0 \end{cases} \tag{(Q)}$$

[7]:
```
#solving the given problem for \vartheta'(G)
P = cp.Variable((n,n), symmetric = True)
k = cp.Variable(1)
```

```
constraints = [P >> 0]
for i in range(n):
    for j in range(i,n):
        constraints.append(k*(I[i][j]+G[i][j]) >= J[i][j] + P[i][j])
#constraints
prob = cp.Problem(cp.Minimize(k), constraints)
print(prob.solve(),"\n")
```

```
3.999999962758152
```

We showed that optval(Q) is equal to

$$\begin{aligned} \min_{\substack{X\in S^n \\ k\in\mathbb{R}}} \quad & k \\ \text{s.t.} \quad & kA \geq X \\ & kI + X \succeq J \end{aligned} \tag{(R)}$$

For sanity check, we also solve this problem and check.

[8]:
```
#solving the given problem for \vartheta'(G)
M = cp.Variable((n,n), symmetric = True)
l = cp.Variable(1)
constraints = [l*I+M >> J]
for i in range(n):
    for j in range(i,n):
        constraints.append(l*G[i][j] >= M[i][j])
#constraints
prob = cp.Problem(cp.Minimize(l), constraints)
print(prob.solve(),"\n")
```

```
3.9999999331124707
```

Problem 3

Show that the following decision problem is NP-complete. **CONCAVE-BOX-QP**: Given a symmetric matrix $Q \in \mathbb{Q}^{n\times n}$, with $Q \preceq 0$, vectors $c, l, u \in \mathbb{Q}^n$, and a scalar $k \in \mathbb{Q}$, decide whether the optimal value of the following optimization problem is less than or equal to k:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \; & x^\top Qx + c^\top x \\ \text{s.t.} \; & -l_i \le x_i \le u_i \; \forall i \in [n]. \end{aligned} \tag{1}$$

Solution

Let's set up some notation. We have the box $B \coloneqq \prod_{i=1}^n [-l_i, u_i]$ with the function $f(x) \coloneqq x^\top Qx + c^\top x$. We want to minimize f over B. Since B is compact, there is an optimal solution x^* with optimal value $f^* = f(x^*)$.

We are asking whether $f(x^*) = f^* \le k$.
This happens iff $\exists x \in B$ such that $f(x) \le k$.

Lemma 4
Consider the function f above defined over B such that Q is negative-definite. Let $a \in B$ be a point for which there is a coordinate i such that $a_i \in (-l_i, u_i)$. In other words, a is not on a corner of B. Then there is a point $x \ne a$ in B such that $f(x) < f(a)$.

Proof. Let f, B, Q, a, i be as in the hypothesis. Then $\exists \varepsilon > 0$ such that $(a_i - 2\varepsilon, a_i + 2\varepsilon) \subset (-l_i, u_i)$. Then note that $Q_{ii} = e_i^\top Q e_i < 0$ because $Q \prec 0$, whence $\varepsilon Q_{ii} < 0$. But one of $(+e_i)^\top(c + 2Qa)$ or $(-e_i)^\top(c + 2Qa)$ must be non-positive. Let e be $+e_i$ or $-e_i$ accordingly. That is, $e^\top(c + 2Qa) \le 0$ and $e \in \{\pm e_i\}$. Take $x \coloneqq a + \varepsilon e$. Then the i^{th} coordinate of x is $a_i \pm \varepsilon \in (-l_i, u_i)$ and all other coordinates are same as those of a. Therefore $x \in B$. Furthermore,

$$\begin{aligned} f(x) - f(a) &= (a + \varepsilon e)^\top Q(a + \varepsilon e) - a^\top Qa + (a + \varepsilon e)^\top c - a^\top c \\ &= 2\varepsilon e^\top Qa + \varepsilon^2 e^\top Qe^\top + \varepsilon e^\top c \\ &= 2\varepsilon e^\top Qa + \varepsilon^2 e_i^\top Qe_i^\top + \varepsilon e^\top c \\ &= 2\varepsilon e^\top Qa + \varepsilon^2 Q_{ii} + \varepsilon e^\top c \\ &= \varepsilon \left(\varepsilon Q_{ii} + 2e^\top Qa + e^\top c\right) \\ &\le \varepsilon^2 Q_{ii} < 0 \\ \implies f(x) < f(a). \end{aligned}$$

■

Recall that the **MAXCUT** problem takes input a graph $G(V, E)$ along with a positive integer k, and answers whether there is a partition $V = V_1 \cup V_2$ such that $\text{cut}(V_1, V_2) \ge k$. Here cut is the number of edges such that the endpoints of the edges don't lie in the same partition.

Lemma 5
Let $G = (V = [n], E)$ be a graph and $V = V_1 \cup V_2$ be a cut. Assign numbers x_i to each vertex given

by $x_i = \begin{cases} 1 & \text{if } i \in V_1 \\ -1 & \text{if } i \in V_2 \end{cases}$. Then $\text{cut}(V_1, V_2) = \frac{1}{2}\left(|E| - \sum_{\{i,j\} \in E} x_i x_j\right)$. This does not change if we reverse the sign of each x_i.

Proof. Among the edges in E, let C be the set of cut-edges and $N = E \setminus C$. So $\sum_{\{i,j\} \in E} x_i x_j = \sum_{\{i,j\} \in C} x_i x_j + \sum_{\{i,j\} \in N} x_i x_j = -|C| + |N| = -|C| + |E| - |C| \implies 2|C| = |E| - \sum_{\{i,j\} \in E} x_i x_j$.
This remains unaffected on reversal of signs because only quadratic forms were involved in the above calculation. ■

Now we give a reduction **MAXCUT** $\longrightarrow$ **CONCAVE-BOX-QP**. Let $(G(V = [n], E), t)$ be an instance of **MAXCUT**, where $t > 0$ is a positive integer. Let λ be large enough so that $A - \lambda I \prec 0$ (reducing eigenvalues of A by λ) where A is the adjacency matrix of G. Consider the input $(Q, c, l, u, k) = (A - \lambda I, 0, \mathbf{1}, \mathbf{1}, 2|E| - 4t - n\lambda)$ to **CONCAVE-BOX-QP**, where $\mathbf{1} = \sum_{i=1}^{n} e_i \in \mathbb{Q}^n$. We are thus interested in asking whether $\min_{x \in [-1,1]^n} x^\top (A - \lambda I) x \leq k$. Suppose minima of this optimization problem occurs at $\alpha \in [-1, 1]^n$. By Lemma 4, each $\alpha_i \in \{\pm 1\}$.

Claim 6
$\exists x \in \{-1, 1\}^n$ such that $x^\top Q x \leq k \iff G$ has a cut of size $\geq t$.

Proof. We first note that $x^\top (A - \lambda I) x = 2 \sum_{\{i,j\} \in E} x_i x_j - \lambda \sum_{i \in [n]} x_i^2$. And if each $x_i \in \{\pm 1\}$ then this quantity is $= 2 \sum_{\{i,j\} \in E} x_i x_j - n\lambda$.

Suppose $x^\top Q x \leq k$ for some $x \in \{-1, 1\}^n$. Take $V_1 = \{i \in V \mid x_i = +1\}, V_2 = V \setminus V_1$. This is a valid partition. The corresponding cut size is $\frac{1}{2}\left(|E| - \sum_{\{i,j\} \in E} x_i x_j\right) = \frac{1}{2}\left(|E| - \frac{1}{2} x^\top Q x + \frac{1}{2} n\lambda\right) \geq \frac{|E|}{2} + \frac{n\lambda}{4} - \frac{k}{4} = t$.

Conversely say $V = V_1 \cup V_2$ is a partition that gives a cut of size $\geq t$. Then take $x_i = \begin{cases} 1 & \text{if } i \in V_1 \\ -1 & \text{if } i \in V_2 \end{cases}$.
So $x^\top (A - \lambda I) x = 2 \sum_{\{i,j\} \in E} x_i x_j - n\lambda = 2|E| - 4\text{cut}(V_1, V_2) - n\lambda \leq 2|E| - 4t - n\lambda = k$. ■

Recall that α is the minimizer of $x^\top Q x$ on $[-1, 1]^n$. So $\min_{x \in [-1,1]^n} x^\top Q x \leq k$ iff $\alpha^\top Q \alpha \leq k$ iff $\exists x \in \{-1, 1\}^n$ such that $x^\top Q x \leq k$. This, along with the above claim proves that $\min_{x \in [-1,1]^n} x^\top Q x \leq k$ iff G has a cut of size $\geq t$. Therefore, this was a valid reduction. To show that this is a poly-time reduction, it is enough to show that λ can be calculated in polynomial time in the input size, which is indeed the case, for example take $\lambda > \|A\|_F = \sqrt{\text{Tr}(A^2)} \geq \lambda_{\max}(A)$. This proves that **CONCAVE-BOX-QP** is NP-hard.

We now show that **CONCAVE-BOX-QP** is in NP. Indeed if we are given a solution $x^* \in \prod_{i=1}^{n}[-l_i, u_i]$ which solves a given instance (Q, c, l, u, k) of the problem, we can simply compute $x^{*\top} Q x^* + c^\top x$ and check whether the value is $\leq k$, and verify whether $x_i^* \in [-l_i, u_i]$ for each $i \in [n]$. The number of operations to find this correctly takes polynomial time in the size of the input, and similarly for the verification of feasibility.

Problem 4

A polynomial $p : \mathbb{R}^n \to \mathbb{R}$ is *separable* if it can be written as $p(x) = \sum_{i=1}^n q_i(x_i)$, where each q_i is a univariate polynomial.

(a) Show that a separable polynomial is nonnegative if and only if it is a sum of squares. (You can use the fact that a univariate nonnegative polynomial is a sum of squares without proof.)

(b) Present an explicit family of degree-4 polynomials $p_n : \mathbb{R}^n \to \mathbb{R}$ such that

  (i) the number of nonglobal local minima of p_n grows exponentially with n,

  (ii) for all n, we have

$$\left[\min_{x\in\mathbb{R}^n} p(x)\right] = \begin{bmatrix} \max_{\gamma\in\mathbb{R}} \gamma \\ \text{s.t. } p_n(x) - \gamma \text{ is a sum of squares} \end{bmatrix}$$

Solution

(a) **First failed attempt (not to be graded)**: Noted that each q_i is bounded below. Take a uniform lower bound $-b$ which works for all q_i and write $p = \sum(q_i + b) - nb$. But we need this extra term $-nb$ to be non-negative. That is, we can increase q_i's as much as we want, but we shouldn't do it by much because then the extra term is negative. Thus one needs them to be tight. So 'increase' each q_i only as much as needed.

Successful attempt: Let $p : \mathbb{R}^n \to \mathbb{R}$ be a separable polynomial with $p(x) = \sum_{i=1}^n q_i(x_i)$.
It is clear that if p is a sum of squares, then it is non-negative.
Now suppose p is non-negative everywhere. We claim that each q_i is bounded below. Indeed if, say WLOG, q_1 is unbounded below, then $\exists t \in \mathbb{R}$ such that $p(t, 0, \cdots, 0) = q_1(t) + \sum_{i=1}^n q_i(0) < 0$. (In particular, each q_i is an even-degree polynomial.) So each of their minima is attained. Say q_i attains minima at t_i and call the optimal values $q_i^* = q_i(t_i)$. Now consider rewriting $p(x) = \sum_{i=1}^n (\underbrace{q_i(x_i) - q_i^*}_{r_i(x_i)}) + \underbrace{\sum_i q_i^*}_{k}$. Clearly $0 \le p(t_1, \cdots, t_n) = k$. But by construction, each $r_i(x_i)$ is non-negative everywhere. Thus each r_i is a sum of squares because it's univariate. So we've written p as a sum of squares, namely the squares coming from each r_i and $\left(\sqrt{k}\right)^2$.

(b) Let's start with the polynomial $f(t) := 12t^3 - 48t^2 + 36t = 12t(t-1)(t-3)$ so that it is the anti-derivtive of $\boxed{q(t) := 3t^4 - 16t^3 + 18t^2}$. So $q'(t) = f(t)$ and $q''(t) = 12(3t^2 - 8t + 3)$

Consider the polynomial $p_n(x_1, \cdots, x_n) := \sum_{i=1}^n q(x_i)$.

<u>Critical points</u>. FONC gives that local minima $\implies$ critical point. We first find points $\overline{x} \in \mathbb{R}^n$ such that $\nabla p_n(\overline{x}) = 0$. Denote $\partial_i := \frac{\partial}{\partial x_i}$. Then $0 = \partial_i p_n(\overline{x}) = \partial_i q(\overline{x}_i) = f(\overline{x}_i) \implies \overline{x}_i \in$

$\{0, 1, 3\}$. Conversely if $\overline{x} \in \mathbb{R}^n$ is such that each coordinate is in $\{0, 1, 3\}$ then clearly the gradient vanishes. Any local minima of p_n is among these 3^n points. So we can now restrict our attention only to points of the mentioned form.

Local minima. SONC gives that local minima $\implies$ PSD hessian. SOSC gives that PD hessian + critical point $\implies$ strict local minima. So a necessary condition is $\nabla^2 p_n(\overline{x}) \succeq 0$. However note that the $(i, j)^{\text{th}}$ entry of the Hessian is $\partial_i \partial_j p_n(\overline{x}) = \partial_i f(\overline{x}_j) = \delta_i^j f'(\overline{x}_i)$. In other words, the Hessian is a diagonal matrix with entries $f'(\overline{x}_i) = 12(3\overline{x}_i^2 - 8\overline{x}_i + 3)$. For the Hessian to be PSD, we require $f'(\overline{x}_i) \geq 0 \forall i \in [n]$. If some critical point $\overline{x}$ has a coordinate (say first, WLOG) equal to 1 then $f'(\overline{x}_1) = 12(3 - 8 + 3) < 0$. So if a critical point is a local minima, all its coordinates must be either 0 or 3 (by SONC). In fact, $f'(0) = 36 > 0$ and $f'(3) = 72 > 0$. By SOSC, the set of points of (strict) local minima of p is $L = \{0, 3\}^n$. Note $|L| = 2^n$.

Let's prove a claim before the next step.

Claim 7
Let $p_n^* = p_n((3, 3, \cdots, 3))$. Then $p_n(x) \geq p^* \forall x \in \mathbb{R}^n$ and is the unique global minima.

Proof. Recall $q'(t)$ has roots $0, 1, 3$ and $q''(1) < 0, q''(0) > 0, q''(3) > 0$. So the only points of global minima for q can be $0, 3$. $q(0) = 0, q(3) = -27$. So 3 is the unique global minima for q. Let $v \in \mathbb{R}^n \setminus \{(3, \cdots, 3)\}$. So $\{j \in [n] : v_j \neq 3\} \neq \varnothing$. Then $p_n(v) - p_n^* = \sum\limits_{j : v_j \neq 3} q_j(v_j) - q_j(3) > 0$ because it's a non-empty sum comprising positive terms. ■

Non-global local minima. We've found 2^n local minima. Now we eliminate those which take lowest value on p_n among these points. Using the above claim, the only local min which is also a global min is $(3, \cdots, 3)$. Therefore the number of non-global local minima is $\geq 2^n - 1$.

The optimization problem. Note that $\min\limits_{x \in \mathbb{R}^n} p_n(x) = -27n$. Furthermore $p_n(x) - \gamma$ is separable for every $\gamma \in \mathbb{R}$ by construction. It follows that $p_n(x) - \gamma \geq 0$ everywhere iff $p(x) - \gamma$ is a sum of squares. Therefore the second (dual) optimization problem is exactly $\max \{\gamma \in \mathbb{R}^n \mid p_n(x) - \gamma \geq 0 \ \forall x \in \mathbb{R}^n\}$. Note that $\gamma = -27n$ is feasible because it is the global minima of p_n. And $\gamma = -27n + \varepsilon$ is not feasible for any $\varepsilon > 0$ because $p(x) + 27n - \varepsilon$ is not always non-negative, for example at the point $x = (3, \cdots, 3)$ the value of the expression is $-\varepsilon < 0$ Thus the dual problem has optimal value $-27n$.

Therefore a family of quartic polynomials as required by the question is

$$p_n(x_1, \cdots, x_n) = \sum_{i=1}^{n} \left(3x_i^4 - 16x_i^3 + 18x_i^2\right).$$

Not for grading: Any polynomial whose graph looks like the following cartoon picture should suffice for the role of q above.