CSE 291-A: Homework 3

Hao-en Sung [A53204772] (wrangle1005@gmail.com)
Department of Computer Science
University of California, San Diego
San Diego, CA 92092

Exercises from textbook Chapter Three and Four

3.54

(a) Log-concavity of Gaussian cumulative distribution function.

Sol. After some calculation, I can find out the first and second order derivatives of $f(x)$ as follows.

$$f'(x) = \frac{e^{\frac{-x^2}{2}}}{\sqrt{2\pi}}$$
$$f''(x) = \frac{-x \cdot e^{\frac{-x^2}{2}}}{\sqrt{2\pi}}$$

Since $f(x) \geq 0$ and $f''(x) \leq 0$ when $x \geq 0$, I have

$$f''(x) \cdot f(x) \leq 0 \leq f'(x)^2.$$

□

(b)

Sol. It is obvious that for all t and x, the following formulas always hold:

$$(t-x)^2 \geq 0$$
$$t^2 + x^2 \geq 2xt$$
$$\frac{t^2}{2} \geq -\frac{x^2}{2} + xt.$$

□

(c)

Sol. One can derive the conclusion from (b) that

$$\frac{t^2}{2} \geq -\frac{x^2}{2} + xt$$
$$e^{-\frac{t^2}{2}} \leq e^{\frac{x^2}{2} - xt}$$
$$\int_{-\infty}^{x} e^{-\frac{t^2}{2}} \cdot dt \leq e^{\frac{x^2}{2}} \cdot \int_{-\infty}^{x} e^{-xt} \cdot dt.$$

□

(d)

Sol. From (c), I have

$$\begin{aligned}
\int_{-\infty}^{x} e^{-\frac{t^2}{2}} \cdot dt &\le e^{\frac{x^2}{2}} \cdot \int_{-\infty}^{x} e^{-xt} \cdot dt \\
&= e^{\frac{x^2}{2}} \cdot -\frac{1}{x} \cdot e^{-xt}\Big|_{-\infty}^{x} \\
&= -\frac{e^{-\frac{x^2}{2}}}{x} \\
-xe^{-\frac{x^2}{2}} \cdot \int_{-\infty}^{x} e^{-\frac{t^2}{2}} \cdot dt &\le e^{-x^2} \\
f''(x) \cdot f(x) &\le f'(x)^2.
\end{aligned}$$

□

4.1

(a)

Sol. After drawing the figure, I know the covered area is in the first quadrant and to the right of line $2x_1 + x_2 = 1$ and line $x_1 + 3x_2 = 1$.

To find out the minimum of $x_1 + x_2$, I can use a line $x_1 + x_2 = d$ to approach that covered area from the bottom-left. By doing so, we can find out the optimal solution is 1 when $x_1 = 0.4, x_2 = 0.2$. □

(b)

Sol. Using similar approach as (a); however, the covered area is unbounded to the right. Thus, the optimal solution is unbounded below. □

(c)

Sol. The optimal value for (c) is 0, which happens when $x_2 \ge 0$. □

(d)

Sol. One can find out that the $\max\{x_1, x_2\}$ is mirrored to the line $x_1 = x_2$. Thus, the optimal of (d) is $\frac{1}{3}$, which happens when $x_1 = x_2 = \frac{1}{3}$. □

(e)

Sol. After using CVX programming in MATLAB, the optimal of (d) can be calculated as $\frac{1}{2}$, which happens when $x_1 = \frac{1}{2}, x_2 = \frac{1}{6}$. □

4.12 Network flow problem.

Sol. According to the problem description, the optimization problem can be written as:

$$\begin{aligned}
\text{minimize} \quad & C = \sum_{i,j=1}^{n} c_{ij} x_{ij} \\
\text{subject to} \quad & \sum_{j=1}^{n} x_{ji} - \sum_{j=1}^{n} x_{ij} = b_i, \ i = 1, \cdots, n \\
& l_{ij} \le x_{ij} \le u_{ij}.
\end{aligned}$$

□

4.28 Robust quadratic programming.

(a)

Sol. Since $\frac{1}{2}x^\intercal P_i x + q^\intercal x + r$ is convex, the maximum among P_i is also convex.

If I assume the maximum among P_i is P_i^* abd it satisfies $\frac{1}{2}x^\intercal P_i^* x + q^\intercal x + r = t$, I can rewrite the origin optimization function as

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \dfrac{1}{2}x^\intercal P_i x + q^\intercal x + r \leq t,\ i = 1, \cdots, K \\ & Ax \preceq b, \end{array}$$

$\square$

which is a QCQP problem.

(b)

Sol. If I denote $\frac{1}{2}x^\intercal P_0 x + q^\intercal x + r$ as Z and put the original objective function into the inequality formula, I have

$$\begin{aligned} & P_0 - \gamma I \preceq P \preceq P_0 + \gamma I \\ & Z - \frac{1}{2}\gamma I \cdot x^\intercal x \preceq \frac{1}{2}x^\intercal P x + q^\intercal x + r \preceq P_0 - \gamma I \preceq P \preceq Z + \frac{1}{2}\gamma I \cdot x^\intercal x. \end{aligned}$$

Thus, I know the maximum happens when $P = P_0 + \gamma I$, which implies a new optimization problem

$$\begin{array}{ll} \text{minimize} & \dfrac{1}{2}x^\intercal \left(P_0 + \gamma I\right) x + q^\intercal x + r \\ & Ax \preceq b, \end{array}$$

which is a normal QP problem. $\square$

(c)

Sol. I can firstly rewrite the objective function as follows.

$$\begin{aligned} \sup_{P \in \mathcal{E}} \frac{1}{2}x^\intercal P x + q^\intercal x + r &= \sup_{\|u\|_2 \leq 1} \left[\frac{1}{2}x^\intercal \left(P_0 + \sum_{i=1}^{K} P_i u_i\right) x + q^\intercal x + r\right] \\ &= \frac{1}{2}x^\intercal P_0 x + \frac{1}{2} \cdot \sup_{\|u\|_2 \leq 1} \left(\sum_{i=1}^{K} u_i \cdot x^\intercal P_i x\right) + q^\intercal x + r \end{aligned}$$

According to Cauchy–Schwarz inequality, I know that

$$\begin{aligned} \left(\sum_{i=1}^{K} u_i \cdot x^\intercal P_i x\right)^2 &\leq \left(\sum_{i=1}^{K} u_i^2\right) \cdot \left(\sum_{i=1}^{K} \left(x^\intercal P_i x\right)^2\right) \\ &\leq 1 \cdot \left(\sum_{i=1}^{K} \left(x^\intercal P_i x\right)^2\right) \\ \sum_{i=1}^{K} u_i \cdot x^\intercal P_i x &\leq \left(\sum_{i=1}^{K} \left(x^\intercal P_i x\right)^2\right)^{\frac{1}{2}} \\ \sup_{\|u\|_2 \leq 1} \left(\sum_{i=1}^{K} u_i \cdot x^\intercal P_i x\right) &= \left(\sum_{i=1}^{K} \left(x^\intercal P_i x\right)^2\right)^{\frac{1}{2}} \end{aligned}$$

$\square$

Thus, the optimization problem can be rewritten as

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^\intercal P_0 x + \frac{1}{2} \cdot \left(\sum_{i=1}^{K} (x^\intercal P_i x)^2 \right)^{\frac{1}{2}} + q^\intercal x + r \\ & Ax \preceq b, \end{array}$$

which is a normal QP problem.

4.37 Generalized posynomials and geometric programming.

Sol. Firstly, I can transform the original optimization problem into

$$\begin{array}{ll} \text{minimize} & t \\ & \frac{h_0(x)}{t} \leq 1 \\ & h_i(x) \leq 1,\ i = 1, \cdots, m \\ & g_i(x) = 1,\ i = 1, \cdots, p. \end{array}$$

Now I just need to convert each $h_i(x)$, including $h_0^*(x) = \frac{h_0(x)}{t}$, into geometric program (GP). Assume that

$$h_i(x) = \phi_i(f_1(x), \cdots, f_k(x)) \leq 1,$$

one can introduce k parameters here and transfer the original generalized posynomial inequality into

$$\begin{array}{ll} h_i(x) = \phi_i(t_1, \cdots, t_k) \leq 1, & \text{s.t.} \\ f_1(x) \leq t_1 & \\ \quad\vdots & \\ f_k(x) \leq t_k. & \end{array}$$

At the end, $1 + k \cdot m$ new variables are introduced to convert original problem from GGP to GP. The proof for equality between two problems are straightforward, since only variable substituions are performed. □

Assignments

[Maximum Flow Problem]

Sol. I simply follow the hints and write down the objective function and all the constraints in CVX format, which can be found in the appendix. After executing the code, CVX solver can easily find out one solution with optimal value equals to 12.

It is interesting to notice that there is some constraints in variable declaration. For example, *variable cd;* will incur unknown error in CVX program. □

Appendix

Listing 1: Code for Maximum Flow using CVX

```
cvx_begin
    variable vsa(1);
    variable vsc(1);
    variable vab(1);
    variable vad(1);
    variable vcd(1);
    variable vcf(1);
    variable vbe(1);
    variable vde(1);
    variable vdf(1);
    variable vfe(1);
    variable vet(1);
    variable vft(1);

    maximize (vsa + vsc);
    subject to
        0 <= vsa <= 6;
        0 <= vsc <= 7;
        0 <= vab <= 5;
        0 <= vad <= 4;
        0 <= vcd <= 1;
        0 <= vcf <= 5;
        0 <= vbe <= 7;
        0 <= vde <= 3;
        0 <= vdf <= 3;
        0 <= vfe <= 2;
        0 <= vet <= 9;
        0 <= vft <= 4;
        vsa <= vab + vad; % a
        vab == vbe; % b
        vsc == vcd + vcf; % c
        vad + vcd == vde + vdf; % d
        vbe + vde + vfe == vet; % e
        vcf + vdf == vfe + vft; % f
cvx_end
```