Homework assignment 1

- Please submit your answers via Gradescope on the CCLE course website. The deadline is Friday 1/15/2021 at 11:59PM.
- Exercise numbers with prefix T refer to the textbook. Exercise numbers with prefix A refer to the collection of additional exercises in the Homework section of the CCLE website.

1. Exercise T2.12 (d, e, g).
2. Exercise A1.6 (a,b,d).
3. Exercise A1.10 (a).
4. *Schur complements and positive semidefinite matrices.* Let X be a symmetric matrix partitioned as

$$X = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}. \tag{1}$$

If A is nonsingular, the matrix $S = C - B^T A^{-1} B$ is called the *Schur complement* of A in X. It A is positive definite, then it can be shown that $X \succeq 0$ (X is positive semidefinite) if and only if $S \succeq 0$ (see page 650 of the textbook). In this exercise we prove the extension of this result to singular A mentioned on page 651 of the textbook.

(a) Suppose $A = 0$ in (1). Show that $X \succeq 0$ if and only if $B = 0$ and $C \succeq 0$.

(b) Let A be a symmetric $n \times n$ matrix with eigendecomposition

$$A = Q \Lambda Q^T,$$

where Q is orthogonal ($Q^T Q = QQ^T = I$) and $\Lambda = \mathbf{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$. Assume the first r eigenvalues λ_i are nonzero and $\lambda_{r+1} = \cdots = \lambda_n = 0$. Partition Q and Λ as

$$Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}, \qquad \Lambda = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & 0 \end{bmatrix}$$

with Q_1 of size $n \times r$, Q_2 of size $n \times (n-r)$, and $\Lambda_1 = \mathbf{diag}(\lambda_1, \ldots, \lambda_r)$. The matrix

$$A^\dagger = Q_1 \Lambda_1^{-1} Q_1^T$$

is called the *pseudo-inverse* of A. Verify that

$$AA^\dagger = A^\dagger A = Q_1 Q_1^T, \qquad I - AA^\dagger = I - A^\dagger A = Q_2 Q_2^T.$$

The matrix–vector product $AA^\dagger x = Q_1 Q_1^T x$ is the orthogonal projection of the vector x on the range of A. The matrix–vector product $(I - AA^\dagger)x = Q_2 Q_2^T x$ is the orthogonal projection on the nullspace of A.

(c) Show that the block matrix X in (1) is positive semidefinite if and only if

$$A \succeq 0, \qquad (I - AA^\dagger)B = 0, \qquad C - B^T A^\dagger B \succeq 0.$$

(The second condition means that the columns of B are in the range of A.)
Hint. Let $A = Q\Lambda Q^T$ be the eigenvalue decomposition of A. Partition Q and Λ as in part (b). The matrix X in (1) is positive semidefinite if and only if the matrix

$$\begin{bmatrix} Q^T & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \begin{bmatrix} Q & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} \Lambda & Q^T B \\ B^T Q & C \end{bmatrix} = \begin{bmatrix} \Lambda_1 & 0 & Q_1^T B \\ 0 & 0 & Q_2^T B \\ B^T Q_1 & B^T Q_2 & C \end{bmatrix}$$

is positive semidefinite. Using the observation in part (a) we see that this matrix is positive semidefinite if and only if $Q_2^T B = 0$ and the matrix

$$\begin{bmatrix} \Lambda_1 & Q_1^T B \\ B^T Q_1 & C \end{bmatrix}$$

is positive semidefinite. Apply the Schur complement characterization for 2×2 block matrices with a positive definite 1,1 block (page 650 of the textbook) to show the result.

5. This problem is an introduction to the MATLAB software package CVX (`cvxr.com`) that will be used in the course[1]. Data files can be found on the CCLE course website.

   We consider the illumination problem of lecture 1. We take $I_{\rm des} = 1$ and $p_{\max} = 1$, so the problem is

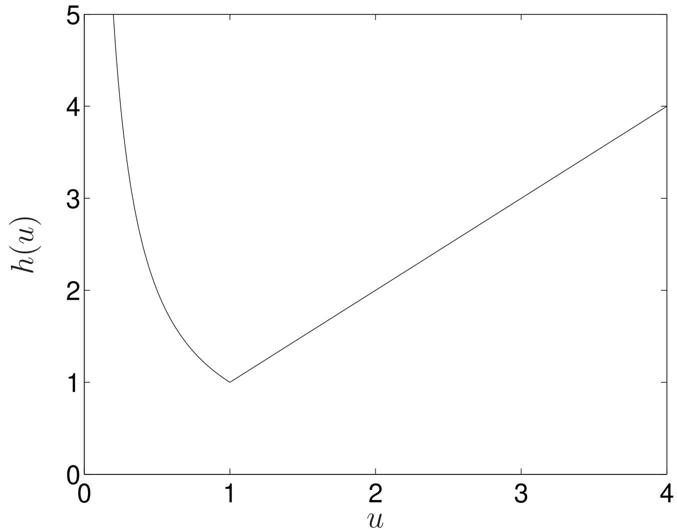
$$\begin{array}{ll} \text{minimize} & f_0(p) = \max\limits_{k=1,\ldots,n} |\log(a_k^T p)| \\ \text{subject to} & 0 \le p_j \le 1, \quad j = 1, \ldots, m, \end{array} \tag{2}$$

   with variable $p \in \mathbf{R}^m$. As mentioned in the lecture, the problem is equivalent to

$$\begin{array}{ll} \text{minimize} & \max\limits_{k=1,\ldots,n} h(a_k^T p) \\ \text{subject to} & 0 \le p_j \le 1, \quad j = 1, \ldots, m, \end{array} \tag{3}$$

   where $h(u) = \max\{u, 1/u\}$ for $u > 0$. The function h, shown in the figure below, is nonlinear, nondifferentiable, and convex.

[1] Python users are welcome to use CVXPY (`cvxpy.org`). Although the problem assignment is written for MATLAB and CVX, the modifications should be straightforward if you are familiar with Python.



To see the equivalence between (2) and (3), we note that

$$\begin{aligned} f_0(p) &= \max_{k=1,\ldots,n} |\log(a_k^T p)| \\ &= \max_{k=1,\ldots,n} \max\{\log(a_k^T p), \log(1/a_k^T p)\} \\ &= \log \max_{k=1,\ldots,n} \max\{a_k^T p, 1/a_k^T p\} \\ &= \log \max_{k=1,\ldots,n} h(a_k^T p), \end{aligned}$$

and since the logarithm is a monotonically increasing function, minimizing f_0 is equivalent to minimizing $\max_{k=1,\ldots,n} h(a_k^T p)$.

The problem data are given in the file `illum_data.m` posted on the course website. Executing this file in MATLAB creates the $n \times m$-matrix A (which has rows a_k^T). There are 10 lamps ($m = 10$) and 20 patches ($n = 20$).

Use the following methods to compute three approximate solutions and the exact solution, and compare the answers (the vectors p and the corresponding values of $f_0(p)$).

(a) *Least-squares with saturation.* Solve the least squares problem

$$\text{minimize} \quad \sum_{k=1}^{n} (a_k^T p - 1)^2 = \|Ap - \mathbf{1}\|_2^2.$$

If the solution has negative coefficients, set them to zero; if some coefficients are greater than 1, set them to 1. Use the MATLAB command `x = A \ b` to solve a least squares problem (minimize $\|Ax - b\|_2^2$).

(b) *Regularized least squares.* Solve the regularized least squares problem

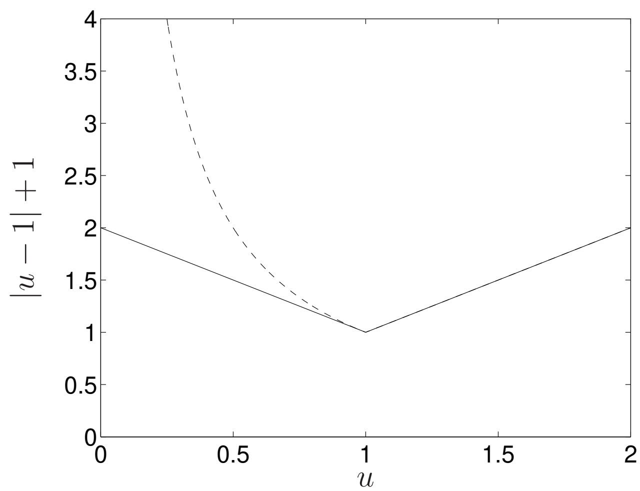
$$\text{minimize} \quad \sum_{k=1}^{n}(a_k^T p - 1)^2 + \rho \sum_{j=1}^{m}(p_j - 0.5)^2 = \|Ap - \mathbf{1}\|_2^2 + \rho\,\|p - (1/2)\mathbf{1}\|_2^2\,,$$

where $\rho > 0$ is a parameter. Increase ρ until all coefficients of p are in the interval $[0, 1]$.

(c) *Chebyshev approximation.* Solve the problem

$$\begin{array}{ll} \text{minimize} & \max\limits_{k=1,\ldots,n} |a_k^T p - 1| = \|Ap - \mathbf{1}\|_\infty \\ \text{subject to} & 0 \le p_j \le 1, \quad j = 1, \ldots, m. \end{array}$$

We can think of this problem as obtained by approximating the nonlinear function $h(u)$ by a piecewise-linear function $|u - 1| + 1$. As shown in the figure below, this is a good approximation around $u = 1$. This problem can be converted to a linear program and solved using the MATLAB function `linprog`. It can also be solved directly in CVX, using the expression `norm(A*p - 1, inf)` to specify the cost function.



(d) *Exact solution.* Finally, use CVX to solve

$$\begin{array}{ll} \text{minimize} & \max\limits_{k=1,\ldots,n} \max(a_k^T p, 1/a_k^T p) \\ \text{subject to} & 0 \le p_j \le 1, \quad j = 1, \ldots, m. \end{array}$$

Use the CVX function `inv_pos()` to expresss the function $f(x) = 1/x$ with domain $\mathbf{R}_{++}$.