Homework 1

- Please submit your answers via Gradescope on the Bruin Learn course website. The deadline is Monday 1/22/2024 at 11:59PM.
- Exercise numbers with prefix T refer to the textbook. Exercise numbers with prefix A refer to the additional exercises under `Files/Homework` on the Bruin Learn site.
- Data files can be found under `Files/Homework/Data files` on the Bruin Learn site.

1. Exercise T2.9 (a).
2. Exercise T2.12 (d, e, g).
3. Exercise A2.10.
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 ($Q^TQ = QQ^T = I$) 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 software packages CVX (`cvxr.com`) and CVXPY (`cvxpy.org`).

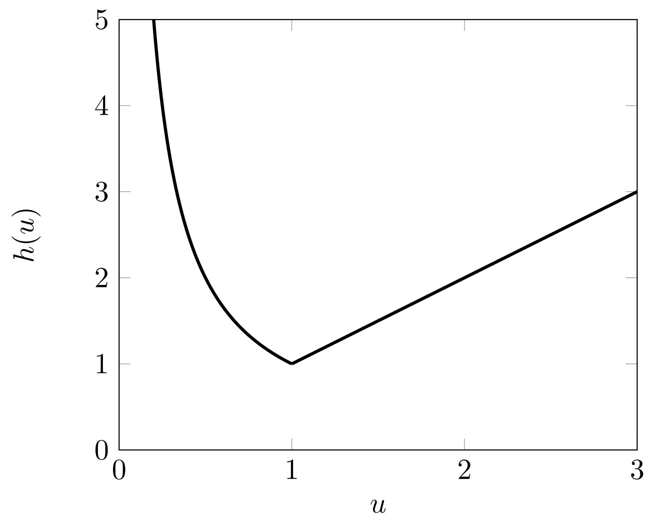
We consider the illumination problem of lecture 1. We take $I_{\rm des} = 1$ and $p_{\max} = 1$, so the problem is

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

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

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

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



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

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

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

We consider a small example with $n = 10$ lamps and $m = 20$ patches. The $m \times n$ matrix A with rows a_k^T is given in the files `illum_data.m` and `illum_data.py` on the course website (in the folder `Files/Homework/Data files`).

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

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

$$\text{minimize} \quad \sum_{k=1}^{m} (a_k^T x - 1)^2 = \|Ax - \mathbf{1}\|_2^2.$$

If the solution has negative coefficients, set them to zero; if some coefficients are greater than 1, set them to 1.

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

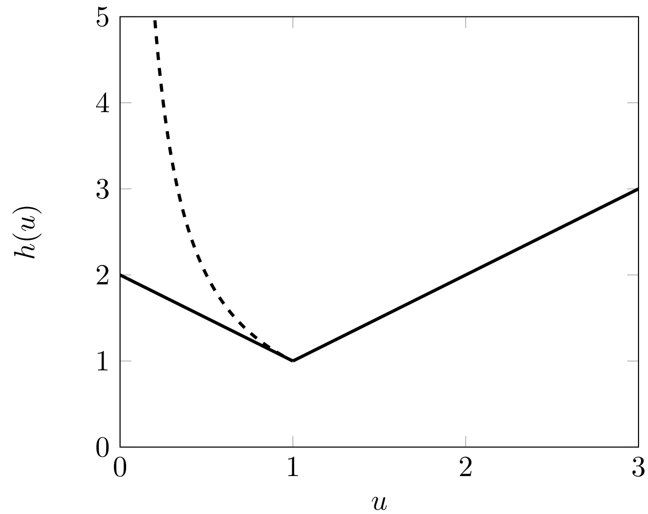
$$\text{minimize} \quad \sum_{k=1}^{m} (a_k^T x - 1)^2 + \rho \sum_{j=1}^{n} (x_j - 0.5)^2 = \|Ax - \mathbf{1}\|_2^2 + \rho \|x - (1/2)\mathbf{1}\|_2^2,$$

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

(c) *Chebyshev approximation.* Solve the problem

$$\begin{array}{ll} \text{minimize} & \displaystyle\max_{k=1,\ldots,m} |a_k^T x - 1| = \|Ax - \mathbf{1}\|_\infty \\ \text{subject to} & 0 \leq x_j \leq 1, \quad j = 1, \ldots, n. \end{array}$$

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 $u = 1$.



(d) *Exact solution.* Solve

$$\begin{array}{ll} \text{minimize} & \displaystyle\max_{k=1,\ldots,m} \max(a_k^T x, 1/a_k^T x) \\ \text{subject to} & 0 \leq x_j \leq 1, \quad j = 1, \ldots, n. \end{array}$$

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