EE364a Homework 3 additional problems

1. *Circularly symmetric Huber function.* The scalar Huber function is defined as

$$f_{\rm hub}(x) = \begin{cases} (1/2)x^2 & |x| \leq 1 \\ |x| - 1/2 & |x| > 1. \end{cases}$$

This convex function comes up in several applications, including robust estimation. This problem concerns generalizations of the Huber function to $\mathbf{R}^n$. One generalization to $\mathbf{R}^n$ is given by $f_{\rm hub}(x_1) + \cdots + f_{\rm hub}(x_n)$, but this function is not circularly symmetric, *i.e.*, invariant under transformation of x by an orthogonal matrix. A generalization to $\mathbf{R}^n$ that *is* circularly symmetric is

$$f_{\rm cshub}(x) = f_{\rm hub}(\|x\|_2) = \begin{cases} (1/2)\|x\|_2^2 & \|x\|_2 \leq 1 \\ \|x\|_2 - 1/2 & \|x\|_2 > 1. \end{cases}$$

(The subscript stands for 'circularly symmetric Huber function'.) Show that $f_{\rm cshub}$ is convex. Find the conjugate function $f^*_{\rm cshub}$.

2. *Minimizing a function over the probability simplex.* Find simple necessary and sufficient conditions for $x \in \mathbf{R}^n$ to minimize a differentiable convex function f over the probability simplex, $\{x \mid \mathbf{1}^T x = 1,\ x \succeq 0\}$.

3. *Reformulating constraints in* `cvx`*.* Each of the following `cvx` code fragments describes a convex constraint on the scalar variables `x`, `y`, and `z`, but violates the `cvx` rule set, and so is invalid. Briefly explain why each fragment is invalid. Then, rewrite each one in an equivalent form that conforms to the `cvx` rule set. In your reformulations, you can use linear equality and inequality constraints, and inequalities constructed using `cvx` functions. You can also introduce additional variables, or use LMIs. Be sure to explain (briefly) why your reformulation is equivalent to the original constraint, if it is not obvious.

Check your reformulations by creating a small problem that includes these constraints, and solving it using `cvx`. Your test problem doesn't have to be feasible; it's enough to verify that `cvx` processes your constraints without error.

Remark. This *looks* like a problem about 'how to use `cvx` software', or 'tricks for using `cvx`'. But it really checks whether you understand the various composition rules, convex analysis, and constraint reformulation rules.

(a) `norm([x + 2*y, x - y]) == 0`

(b) `square(square(x + y)) <= x - y`

(c) `1/x + 1/y <= 1; x >= 0; y >= 0`

(d) `norm([max(x,1), max(y,2)]) <= 3*x + y`

(e) `x*y >= 1; x >= 0; y >= 0`

(f) `(x + y)^2/sqrt(y) <= x - y + 5`

(g) `x^3 + y^3 <= 1; x >= 0; y >= 0`

(h) `x + z <= 1 + sqrt(x*y - z^2); x >= 0; y >= 0`

4. *Optimal activity levels.* Solve the optimal activity level problem described in exercise 4.17 in *Convex Optimization*, for the instance with problem data

$$A = \begin{bmatrix} 1 & 2 & 0 & 1 \\ 0 & 0 & 3 & 1 \\ 0 & 3 & 1 & 1 \\ 2 & 1 & 2 & 5 \\ 1 & 0 & 3 & 2 \end{bmatrix}, \quad c^{\max} = \begin{bmatrix} 100 \\ 100 \\ 100 \\ 100 \\ 100 \end{bmatrix}, \quad p = \begin{bmatrix} 3 \\ 2 \\ 7 \\ 6 \end{bmatrix}, \quad p^{\rm disc} = \begin{bmatrix} 2 \\ 1 \\ 4 \\ 2 \end{bmatrix}, \quad q = \begin{bmatrix} 4 \\ 10 \\ 5 \\ 10 \end{bmatrix}.$$

You can do this by forming the LP you found in your solution of exercise 4.17, or more directly, using `cvx`. Give the optimal activity levels, the revenue generated by each one, and the total revenue generated by the optimal solution. Also, give the average price per unit for each activity level, *i.e.*, the ratio of the revenue associated with an activity, to the activity level. (These numbers should be between the basic and discounted prices for each activity.) Give a *very brief* story explaining, or at least commenting on, the solution you find.

5. *The illumination problem.* This exercise concerns the illumination problem described in lecture 1 (pages 9–11). We'll take $I_{\rm des} = 1$ and $p_{\max} = 1$, so the problem is

$$\begin{array}{ll} \mbox{minimize} & f_0(p) = \max_{k=1,\ldots,n} |\log(a_k^T p)| \\ \mbox{subject to} & 0 \leq p_j \leq 1, \quad j = 1, \ldots, m, \end{array} \qquad (1)$$

with variable $p \in \mathbf{R}^m$. You will compute several approximate solutions, and compare the results to the exact solution, for a specific problem instance.

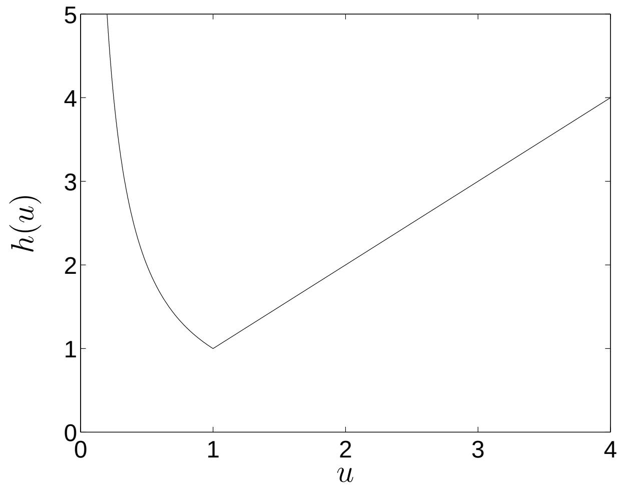
As mentioned in the lecture, the problem is equivalent to

$$\begin{array}{ll} \mbox{minimize} & \max_{k=1,\ldots,n} h(a_k^T p) \\ \mbox{subject to} & 0 \leq p_j \leq 1, \quad j = 1, \ldots, m, \end{array} \qquad (2)$$

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 (1) and (2), we note that

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

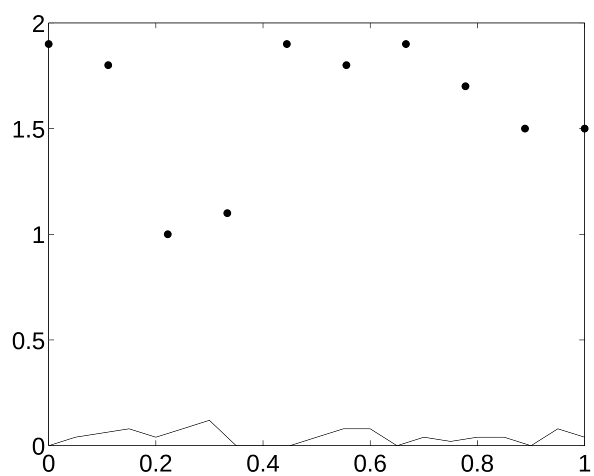
and since the logarithm is a monotonically increasing function, minimizing f_0 is equivalent to minimizing $\max_{k=1,\ldots,n} h(a_k^T p)$.



The problem instance. The specific problem data are for the geometry shown below, using the formula

$$a_{kj} = r_{kj}^{-2} \max\{\cos\theta_{kj}, 0\}$$

from the lecture. There are 10 lamps ($m = 10$) and 20 patches ($n = 20$). We take $I_{\text{des}} = 1$ and $p_{\max} = 1$. The problem data are given in the file `illum_data.m` on the course website. Running this script will construct the matrix A (which has rows a_k^T), and plot the lamp/patch geometry as shown below.



Equal lamp powers. Take $p_j = \gamma$ for $j = 1, \ldots, m$. Plot $f_0(p)$ versus γ over the interval $[0, 1]$. Graphically determine the optimal value of γ, and the associated objective value.

You can evaluate the objective function $f_0(p)$ in Matlab as `max(abs(log(A*p)))`.

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 values for some p_i, set them to zero; if some values are greater than 1, set them to 1. Give the resulting value of $f_0(p)$.

Least-squares solutions can be computed using the Matlab backslash operator: `A\b` returns the solution of the least-squares problem

$$\text{minimize} \quad \|Ax - b\|_2^2.$$

Regularized least-squares. Solve the regularized least-squares problem

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

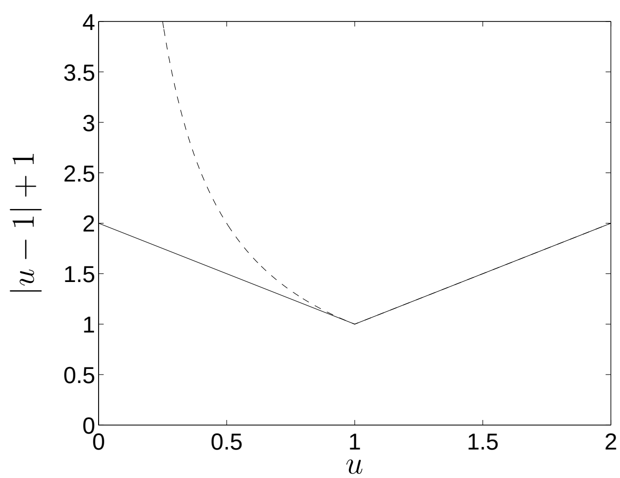
where $\rho > 0$ is a parameter. Increase ρ until all coefficients of p are in the interval $[0, 1]$. Give the resulting value of $f_0(p)$.

You can use the backslash operator in Matlab to solve the regularized least-squares problem.

Chebyshev approximation. Solve the problem

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

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 $u = 1$.



You can solve the Chebyshev approximation problem using `cvx`. The (convex) function $\|Ap - \mathbf{1}\|_\infty$ can be expressed in `cvx` as `norm(A*p-ones(n,1),inf)`. Give the resulting value of $f_0(p)$.

Exact solution. Finally, use `cvx` to solve

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

exactly. You may find the `inv_pos()` function useful. Give the resulting (optimal) value of $f_0(p)$.