# Homework 4

**Due date: Wednesday, November 28.**

Again, you may want to start early.

1. You are given two polyhedra $\mathcal{P}_1$ and $\mathcal{P}_2$ given by $\mathcal{P}_1 := \{x \in \mathbf{R}^n : A_1 x \leq b_1\}$ and $\mathcal{P}_2 := \{x \in \mathbf{R}^n : A_2 x \leq b_2\}$.

   (a) Suppose the polyhedra do not intersect, i.e. $\mathcal{P}_1 \cap \mathcal{P}_2 = \emptyset$, then there exists a separating hyperplane s.t. $a^T x < c$ for every $x \in \mathcal{P}_1$ and $a^T x > c$ for every $x \in \mathcal{P}_2$.
   Describe a method for constructing such a hyperplane. Your method should take $A_1, b_1, A_2, b_2$ as input and return the vector $a$ and $c$ obeying the above relations. (You might want to write down some pseudo-code that explains how you would do it on the computer. However, you do not need to write some formal code.)

   (b) Can you adapt your method to also let the user know when $\mathcal{P}_1 \cap \mathcal{P}_2 \neq \emptyset$?

   (c) *Bonus:* You will get extra credit if you implement this (you are welcome to use `cvx` if this is useful).

2. Boyd and Vandenberghe, Chapter 9, Problem 9.4.

3. Boyd and Vandenberghe, Chapter 4, Problem 9.10.

4. Boyd and Vandenberghe, Chapter 4, Problem 9.13.

5. This problem is attributed to Boyd and the goal is to study the numerical performance of the gradient and Newton methods. You are welcome to use the code I distributed in class.

   Consider the unconstrained problem

   $$\min f(x) = -\sum_{i=1}^{m} \log(1 - a_i^T x) - \sum_{i=1}^{n} \log(1 - x_i^2)$$

   with variable $x \in \mathbf{R}^n$ and

   $$\mathbf{dom} f = \{x : a_i^T x < 1; i = 1, \ldots, m, |x_i| < 1, i = 1, \ldots, n\}.$$

   Note that we can choose $x_0 = 0$ as our initial point. You can generate instances of this problem by choosing $a_i$ at random from some distribution on $\mathbf{R}^n$.

   (a) Use the gradient method to solve the problem using bactracking line search and a stopping criterion of the form $\|\nabla f(x)\| \leq \eta$ . Plot the objective function and step length versus iteration number. (Once you have determined $p^\star$ to high accuracy, you can also plot $f - p^\star$ versus iteration.) Experiment with the backtracking parameters (reasonable choices) to see their effect on the total number of iterations required. Carry these experiments out for several instances of the problem, of different sizes.

   (b) Repeat using Newton's method, with stopping criterion based on the Newton decrement $\lambda^2$. Check the quadratic convergence phase.