CO 602: Fundamentals of Optimization
Fall 2010
Problem Set 2
S. Vavasis

Handed out: 2010-Sep-27.

Due: 2010-Oct-4 in lecture.

1. A sequence of k points in $\mathbf{R}^d$, say, $\mathbf{x}_1, \ldots, \mathbf{x}_k$, is called *affinely independent* if the vectors $[\mathbf{x}_1; 1], [\mathbf{x}_2; 1], \ldots, [\mathbf{x}_k; 1]$ are linearly independent in $\mathbf{R}^{d+1}$. (The Matlab notation $[\mathbf{x}; 1]$ denotes the vector formed by appending a '1' entry at the end of $\mathbf{x}$.) A *simplex* T is the convex hull of k affinely independent points $\mathbf{x}_1, \ldots, \mathbf{x}_k$, all lying in $\mathbf{R}^d$ for some $d \geq k-1$. These k points are the *vertices* of the simplex. (For example, when $k = 3$, the simplex is a triangle.)

   (a) Show that any point in a simplex is uniquely written as a convex combination of its vertices, i.e., given $\mathbf{y} \in T$, there is only one choice of nonnegative $\lambda_1, \ldots, \lambda_k$ adding to 1 such that $\lambda_1 \mathbf{x}_1 + \cdots + \lambda_k \mathbf{x}_k = \mathbf{y}$.

   (b) Show that a point $\mathbf{x} \in T$ is extreme if and only if it is a vertex. [Hint: To show that the vertices are extreme, consider a convex combination $\mathbf{p}$ of $\mathbf{y}, \mathbf{z} \in T$. Write $\mathbf{p}$ as in (a), and argue that the coefficient of $\mathbf{x}_i$ must be less than 1 unless $\mathbf{y} = \mathbf{x}_i$ or $\mathbf{z} = \mathbf{z}_i$. But if $\mathbf{p}$'s coefficient of $\mathbf{x}_i$ is less than 1, then $\mathbf{p} \neq \mathbf{x}_i$. Thus $\mathbf{x}_i$ is extreme. To argue that a nonvertex point $\mathbf{q}$ is not extreme, first show that $\mathbf{q}$ must have at least two positive λ_i's in the expansion in part (a). Using these two positive λ_i's it is possible to construct two points $\mathbf{q}', \mathbf{q}'' \in T$ whose midpoint is $\mathbf{q}$.]

2. The text defines a *vertex* of a convex set S to be a point $\mathbf{x} \in S$ such that there exists a $\mathbf{c}$ such that $\mathbf{c}^T\mathbf{x} < \mathbf{c}^T\mathbf{y}$ for all $\mathbf{y} \in S - \{\mathbf{x}\}$. It defines an *extreme point* to be a point $\mathbf{x} \in S$ such that there do not exist $\mathbf{y}, \mathbf{z} \in S - \{\mathbf{x}\}$ such that $\mathbf{x}$ is a convex combination of $\mathbf{y}, \mathbf{z}$. In lecture it was proved that these definitions coincide for polyhedra. Come up with a counterexample to show that they do not coincide for general convex sets. [Hint: in two dimensions, construct a convex domain by joining a square and a half-disk. The resulting domain will have two extreme points that are not vertices. Remark: for the purpose of answering this question, use the definitions of 'extreme point' and 'vertex' provided herein rather than the definitions stated in Q1.]

3. Referring to PS1, Q4(a), recall that the compressive sensing problem was rewritten in standard equality form. Show that at a BFS $(\mathbf{y}, \mathbf{z})$ of this problem, for all i, either $y(i) = 0$ or $z(i) = 0$.

4. Write a matlab program that converts a two-dimensional polytope from its representation as a convex hull to a representation in the form of inequality constraints. The calling sequence of the function should be as follows.

```
function [A,b] = convertToIneq(x)
% The input is a k-by-2 matrix x of coordinates of points in the plane.
% The output is an m-by-2 matrix A and an m-vector b with the property
% that for an arbitrary 2-vector y,
%   A*y >= b   if and only if y in conv(x(1,:),...,x(k,:))
```

The suggested approach is as follows. Use the `convhull` function built into Matlab to obtain the sequence of points chosen from `x` that lie on the boundary of the convex hull. Although not stated in the Matlab documentation, the order of the points produced by `convhull` is always counterclockwise. Thus, each pair of consecutive points of the hull's boundary according to this output defines exactly one inequality constraint.

In order to assist the marking, please scale the output so that for each $i = 1, \ldots, m$, $\|A(i,:)\| = 1$. There is some test data posted on the course website that uses this scaling to formulate the output.

Hand in: listing of all code, printouts of runs on the test cases on the course website.