EE364b Homework 4

1. *Projection onto the probability simplex.* In this problem you will work out a simple method for finding the Euclidean projection y of $x \in \mathbf{R}^n$ onto the probability simplex $\mathcal{P} = \{z \mid z \succeq 0,\ \mathbf{1}^T z = 1\}$.

   Hints. Consider the problem of minimizing $(1/2)\|y - x\|_2^2$ subject to $y \succeq 0$, $\mathbf{1}^T y = 1$. Form the partial Lagrangian

   $$L(y, \nu) = (1/2)\|y - x\|_2^2 + \nu(\mathbf{1}^T y - 1),$$

   leaving the constraint $y \succeq 0$ implicit. Show that $y = (x - \nu\mathbf{1})_+$ minimizes $L(y, \nu)$ over $y \succeq 0$.

2. *Minimizing expected maximum violation.* We consider the problem of minimizing the expected maximum violation of a set of linear constraints subject to a norm bound on the variable,

   $$\begin{array}{ll} \text{minimize} & \mathbf{E}\max(b - Ax)_+ \\ \text{subject to} & \|x\|_\infty \leq 1, \end{array}$$

   where the data $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$ are random.

   We consider a specific problem instance with $m = 3$ and $n = 3$. The entries of A and b vary uniformly (and independently) ± 0.1 around their expected values,

   $$\mathbf{E}\,A = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1/2 & 0 \\ 1 & 1 & 1/2 \end{bmatrix}, \qquad \mathbf{E}\,b = \begin{bmatrix} 9/10 \\ 1 \\ 9/10 \end{bmatrix}.$$

   (a) *Solution via stochastic subgradient.* Use a stochastic subgradient method with step size $1/k$ to compute a solution x^{stoch}, starting from $x = 0$, with $M = 1$ subgradient sample per iteration. Run the algorithm for 5000 iterations. Estimate the objective value obtained by x^{stoch} using Monte Carlo, with $M = 1000$ samples. Plot the distribution of $\max(b - Ax^{\text{stoch}})$ from these samples. (In this plot, points to the left of 0 correspond to no violation of the inequalities.)

   (b) *Maximum margin heuristic.* The heuristic x^{mm} is obtained by maximizing the margin in the inequalities, with the coefficients set to their expected values:

   $$\begin{array}{ll} \text{minimize} & \max(\mathbf{E}\,b - \mathbf{E}\,Ax) \\ \text{subject to} & \|x\|_\infty \leq 1. \end{array}$$

   Use Monte Carlo with $M = 1000$ samples to estimate the objective value (for the original problem) obtained by x^{mm}, and plot the distribution of $\max(b - Ax^{\text{mm}})$.

(c) *Direct solution of sampled problem.* Generate $M = 100$ samples of A and b, and solve the problem

$$\begin{array}{ll} \text{minimize} & (1/M)\sum_{i=1}^{M} \max(b^i - A^i x)_+ \\ \text{subject to} & \|x\|_\infty \leq 1. \end{array}$$

The solution will be denoted x^{ds}. Use Monte Carlo with $M = 1000$ samples to estimate the objective value (for the original problem) obtained by x^{ds}, and plot the distribution of $\max(b - Ax^{\text{ds}})$.

Hints.

- Use `x = max( min(x,1), -1 )` to project onto the ℓ_∞ norm ball.
- Use the `cvx` function `pos()` to get the positive part function $()_+$.
- The clearest code for carrying out Monte Carlo analysis uses a `for` loop. In Matlab `for` loops can be very slow, since they are interpreted. Our for-loop implementation of the solution to this problem isn't too slow, but if you find Monte Carlo runs slow on your machine, you can use the loop-free method shown below, to find the empirical distribution of $\max(b - Ax)$.

```
% loop-free Monte Carlo with 1000 samples of data A and b
M = 1000; noise = 0.1;
Amtx = repmat(Abar,M,1) + 2*noise*rand(M*m,n) - noise;
bmtx = repmat(bbar,M,1) + 2*noise*rand(M*m,1) - noise;
% evaluate max( b - Ax ) for 1000 samples
fvals_stoch = max( reshape(bmtx - Amtx*x_stoch,m,M) );
```

3. *Subgradient method for inequality form SDP.* Describe how to implement a subgradient method to solve the inequality form SDP

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & x_1 A_1 + \cdots + x_n A_n \preceq B, \end{array}$$

with variable $x \in \mathbf{R}^n$, and problem data $c \in \mathbf{R}^n$, $A_1, \ldots, A_n \in \mathbf{S}^m$, $B \in \mathbf{S}^m$.

Generate a small instance of the problem (say, with $n = 4$ and $m = 10$) and solve it using your subgradient method. Check your solution using CVX.

4. *Kelley's cutting-plane algorithm.* We consider the problem of minimizing a convex function $f : \mathbf{R}^n \to \mathbf{R}$ over some convex set C, assuming we can evaluate $f(x)$ and find a subgradient $g \in \partial f(x)$ for any x. Suppose we have evaluated the function and a subgradient at $x^{(1)}, \ldots, x^{(k)}$. We can form the piecewise-linear approximation

$$\hat{f}^{(k)}(x) = \max_{i=1,\ldots,k} \left(f(x^{(i)}) + g^{(i)T}(x - x^{(i)}) \right),$$

which satisfies $\hat{f}^{(k)}(x) \leq f(x)$ for all x. It follows that

$$L^{(k)} = \inf_{x \in C} \hat{f}^{(k)}(x) \leq p^\star,$$

where $p^\star = \inf_{x\in C} f(x)$. Since $\hat{f}^{(k+1)}(x) \geq \hat{f}^{(k)}(x)$ for all x, we have $L^{(k+1)} \geq L^{(k)}$.

In Kelley's cutting-plane algorithm, we set $x^{(k+1)}$ to be any point that minimizes $\hat{f}^{(k)}$ over $x \in C$. The algorithm can be terminated when $U^{(k)} - L^{(k)} \leq \epsilon$, where $U^{(k)} = \min_{i=1,\ldots,k} f(x^{(i)})$.

Use Kelley's cutting-plane algorithm to minimize the piecewise-linear function $f(x) = \max_{i=1,\ldots,m}(a_i^T x + b_i)$ that we have used for other numerical examples, with C the unit cube, *i.e.*, $C = \{x \mid \|x\|_\infty \leq 1\}$. The data that defines the particular function can be found in the Matlab directory of the subgradient notes on the course web site. You can start with $x^{(1)} = 0$ and run the algorithm for 40 iterations. Plot $f(x^{(k)})$, $U^{(k)}$, $L^{(k)}$ and the constant $p^\star$ (on the same plot) versus k.

Repeat for $f(x) = \|x - c\|_2$, where c is chosen from a uniform distribution over the unit cube C. (The solution to this problem is, of course, $x^\star = c$.)

5. *Minimum volume ellipsoid covering a half-ellipsoid.* In this problem we derive the update formulas used in the ellipsoid method, *i.e.*, we will determine the minimum volume ellipsoid that contains the intersection of the ellipsoid

$$\mathcal{E} = \{x \in \mathbf{R}^n \mid (x - x_c)P^{-1}(x - x_c) \leq 1\}$$

and the halfspace

$$\mathcal{H} = \{x \mid g^T(x - x_c) \leq 0\}.$$

We'll assume that $n > 1$, since for $n = 1$ the problem is easy.

(a) We first consider a special case: $\mathcal{E}$ is a ball centered at the origin ($P = I$, $x_c = 0$), and $g = -e_1$ (e_1 the first unit vector), so $\mathcal{E} \cap \mathcal{H} = \{x \mid x^T x \leq 1,\ x_1 \geq 0\}$.
Let

$$\tilde{\mathcal{E}} = \{x \mid (x - \tilde{x}_c)^T \tilde{P}^{-1}(x - \tilde{x}_c) \leq 1\}$$

denote the minimum volume ellipsoid containing $\mathcal{E} \cap \mathcal{H}$. Since $\mathcal{E} \cap \mathcal{H}$ is symmetric about the line through first unit vector e_1, it is clear (and not too hard to show) that $\tilde{\mathcal{E}}$ will have the same symmetry. This means that the matrix $\tilde{P}$ is diagonal, of the form $\tilde{P} = \mathbf{diag}(\alpha, \beta, \beta, \ldots, \beta)$, and that $x_c = \gamma e_1$.

So now we have only three variables to determine: α, β, and γ. Express the volume of $\tilde{\mathcal{E}}$ in terms of these variables, and also the constraint that $\tilde{\mathcal{E}} \supseteq \mathcal{E} \cap \mathcal{H}$. Then solve the optimization problem directly, to show that

$$\alpha = \frac{n^2}{(n+1)^2}, \quad \beta = \frac{n^2}{n^2 - 1}, \quad \gamma = \frac{1}{n+1}$$

(which agrees with the formulas we gave, for this special case).

(b) Now consider the general case, stated at the beginning of this problem. Show how to reduce the general case to the special case solved in part (a).

Hint. Find an affine transformation that maps the original ellipsoid to the unit ball, and g to $-e_1$. Explain why minimizing the volume in these transformed coordinates also minimizes the volume in the original coordinates.