Étude Problem

This is one of the étude problems for this winter. Please turn it in by uploading to Gradescope by 5pm on Thursday, with a regraded version by 5pm on Friday. We will not be lenient with upload times: because of the nature of the regrading, we will be posting solutions as close as possible to 5:01pm on Thursdays, so turning the étude in late will result in no credit. Please upload a placeholder to Gradescope well before the due date.

You may use any books, notes, or computer programs, but you may not discuss the étude with anyone—including online—until the solutions are posted. The only exception is that you can ask us for clarification, via the course staff email address. We’ve tried pretty hard to make these questions unambiguous and clear, so we’re unlikely to say much.

Please submit your étude via Gradescope.

We will deduct points from long, needlessly complex solutions, even if they are correct. Our solutions are not long, so if you find that your solution to a problem goes on and on for many pages, you should try to figure out a simpler one. We expect neat, legible études from everyone, including those enrolled Cr/N.

When a problem involves computation you must give all of the following: a clear discussion and justification of exactly what you did, the source code that produces the result, and the final numerical results or plots.

Files containing problem data can be found in the following location:

http://www.stanford.edu/~jduchi/teaching/364a/data/

Please respect the honor code. Although we allow you to work on homework assignments in small groups, you cannot discuss the études with anyone.

Some études are (quite) straightforward. Others, not so much.

Be sure you are using the most recent version of CVX, CVXPY, CVXR, or Convex.jl.

Some problems involve applications. But you do not need to know anything about the problem area to solve the problem; the problem statement contains everything you need.
9. **Matrix sensing and sequential convex optimization.** In the matrix sensing problem, we receive observations of the form

\[ b_i = \text{tr} A_i^T X, \quad i = 1, \ldots, m, \]

where \( X \in \mathbb{R}^{n_1 \times n_2} \) is assumed to be low rank, and \( A_i \in \mathbb{R}^{n_1 \times n_2} \) as well. In this problem, we will assume that this (unknown) \( X \) is rank 1, so that we may write \( X = x_\star y_\star^T \), where \( x_\star \in \mathbb{R}^{n_1} \) and \( y_\star \in \mathbb{R}^{n_2} \). To estimate \( X \), we therefore wish to solve the (non-smooth, non-convex) problem

\[ \min_{x \in \mathbb{R}^{n_1}, y \in \mathbb{R}^{n_2}} \sum_{i=1}^m |b_i - \text{tr} A_i^T xy^T| = \sum_{i=1}^m |b_i - x^T A_i y| \]  

(1)

in variables \( x \in \mathbb{R}^{n_1}, y \in \mathbb{R}^{n_2} \). Luckily, this is an instance of a composite optimization problem, which we can (approximately) minimize by sequentially minimizing convex approximations.

The general composite problem is as follows: we wish to minimize \( f(x) = h(c(x)) \), where \( h : \mathbb{R}^m \to \mathbb{R} \) is convex and \( c : \mathbb{R}^n \to \mathbb{R}^m \) is a smooth function. Abusing notation and letting \( \nabla c(x) \in \mathbb{R}^{m \times n} \) denote the matrix \([\nabla c_1(x) \cdots \nabla c_m(x)]\) of the gradients of the component functions making up \( c \), a natural model of \( f \) at a point \( x \) is

\[ f_x(x + \Delta) := h(c(x) + \nabla c(x)^T \Delta), \]

which is evidently convex in \( \Delta \). Then one version of sequential convex optimization iterates via

\[ \Delta x := \arg \min_{\Delta} \left\{ f_x(x^k + \Delta) + \frac{1}{2\alpha} \| \Delta \|^2 \right\}, \]

\[ x^{k+1} := x^k + \Delta x \]

where \( \alpha > 0 \) is a stepsize chosen to guarantee that the objectives \( f(x^k) \) are decreasing. We'll specialize this iteration to the matrix sensing problem (1).

For the rest of the problem, we will assume the sensing matrices \( A_i \) take the form \( A_i = u_i v_i^T \) for vectors \( u_i \in \mathbb{R}^{n_1}, v_i \in \mathbb{R}^{n_2} \), so the observations in the sensing problem now take the form

\[ b_i = x_\star^T u_i y_\star^T v_i, \quad i = 1, \ldots, m. \]

(a) Write problem (1) in the form

\[ f(x, y) = \sum_{i=1}^m h(c_i(x, y)) \]

where \( h : \mathbb{R} \to \mathbb{R}_+ \) is a convex function and \( c_i : \mathbb{R}^{n_1+n_2} \to \mathbb{R} \). Your formulation should satisfy

\[ \nabla c_i(x, y) = \begin{bmatrix} (v_i^T y) u_i^T \\ (u_i^T x) v_i \end{bmatrix}. \]

Define the linear mapping \( \mathcal{A} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m \) by

\[ \mathcal{A}(X) = [\text{tr} A_1^T X \quad \text{tr} A_2^T X \quad \cdots \quad \text{tr} A_m^T X]^T. \]

and define the matrices

\[ U = [u_1 \ u_2 \ \cdots \ u_m]^T \in \mathbb{R}^{m \times n_1} \quad \text{and} \quad V = [v_1 \ v_2 \ \cdots \ v_m]^T \in \mathbb{R}^{n \times n_2}. \]
Then assuming the normalization condition $\|u_i\|_2 = \|v_i\|_2 = 1$ for each $i = 1, \ldots, m$, it is not hard to show that if we define the model

$$f(x, y)(x + \Delta x, y + \Delta y) := \|b - A(xy^T) - \text{diag}(Vy)U\Delta x - \text{diag}(Ux)V\Delta y\|_1,$$

which is centered at $(x, y)$ and has variables $\Delta x \in \mathbb{R}^{n_1}$ and $\Delta y \in \mathbb{R}^{n_2}$, then

$$f(x + \Delta x, y + \Delta y) \leq f(x, y)(x + \Delta x, y + \Delta y) + \frac{1}{2}\Delta x^T U^T U \Delta x + \frac{1}{2}\Delta y^T V^T V \Delta y$$

for any $\Delta x \in \mathbb{R}^{n_1}, \Delta y \in \mathbb{R}^{n_2}$.

(b) Using the bound (2), argue that from any initial points $(x^0, y^0)$, the iteration

$$(\Delta x^k, \Delta y^k) := \arg\min_{\Delta x, \Delta y} \left\{ f(x^k, y^k)(x^k + \Delta x, y^k + \Delta y) + \frac{1}{2}\Delta x^T U^T U \Delta x + \frac{1}{2}\Delta y^T V^T V \Delta y \right\}$$

$$(x^{k+1}, y^{k+1}) := (x^k + \Delta x^k, y^k + \Delta y^k)$$

is a descent (or at least, non-ascent) method, that is, $f(x^{k+1}, y^{k+1}) \leq f(x^k, y^k)$.

(c) Using the data in the file `matrix_sco_data.*`, which defines a $U$ matrix, $V$ matrix, and $b$ vector, implement the procedure from part (b). Have your procedure iterate until the change between iterations satisfies

$$\|\Delta x^k\|_2^2 + \|\Delta y^k\|_2^2 \leq \epsilon^2$$

where $\epsilon = 10^{-4}$.

Run your procedure on 10 different random initializations $x^0, y^0$, choosing the entries of each vector to be independent $\mathcal{N}(0, 1)$ random variables. Plot the error $f(x^k, y^k)$ on a semilog plot (logarithmic vertical axis) over iterations for each of the runs of your method. (All 10 runs should be in the same plot.)