Problem 1 (censored data fitting) – cens_fit_data.m, 2 points (additional exercise A5.13 to Boyd and Vandenberghe).

Fitting with censored data. In some experiments there are two kinds of measurements or data available: The usual ones, in which you get a number (say), and *censored data*, in which you don't get the specific number, but are told something about it, such as a lower bound. A classic example is a study of lifetimes of a set of subjects (say, laboratory mice). For those who have died by the end of data collection, we get the lifetime. For those who have not died by the end of data collection, we do not have the lifetime, but we do have a lower bound, *i.e.*, the length of the study. These are the censored data values.

We wish to fit a set of data points,

$$(x^{(1)}, y^{(1)}), \ldots, (x^{(K)}, y^{(K)}),$$

with $x^{(k)} \in \mathbf{R}^n$ and $y^{(k)} \in \mathbf{R}$, with a linear model of the form $y \approx c^T x$. The vector $c \in \mathbf{R}^n$ is the model parameter, which we want to choose. We will use a least-squares criterion, *i.e.*, choose c to minimize

$$J = \sum_{k=1}^{K} \left(y^{(k)} - c^T x^{(k)} \right)^2.$$

Here is the tricky part: some of the values of $y^{(k)}$ are censored; for these entries, we have only a (given) lower bound. We will re-order the data so that $y^{(1)}, \ldots, y^{(M)}$ are given (*i.e.*, uncensored), while $y^{(M+1)}, \ldots, y^{(K)}$ are all censored, *i.e.*, unknown, but larger than D, a given number. All the values of $x^{(k)}$ are known.

(a) Explain how to find c (the model parameter) and $y^{(M+1)}, \ldots, y^{(K)}$ (the censored data values) that minimize J.

(b) Carry out the method of part (a) on the data values in `cens_fit_data.m`. Report $\hat{c}$, the value of c found using this method.

Also find $\hat{c}_{\rm ls}$, the least-squares estimate of c obtained by simply ignoring the censored data samples, *i.e.*, the least-squares estimate based on the data

$$(x^{(1)}, y^{(1)}), \ldots, (x^{(M)}, y^{(M)}).$$

The data file contains $c_{\rm true}$, the true value of c, in the vector `c_true`. Use this to give the two relative errors

$$\frac{\|c_{\rm true} - \hat{c}\|_2}{\|c_{\rm true}\|_2}, \qquad \frac{\|c_{\rm true} - \hat{c}_{\rm ls}\|_2}{\|c_{\rm true}\|_2}.$$

Problem 2 (vehicle speed scheduling) – veh_speed_sched_data.m, 3 points (additional exercise A3.20 to Boyd and Vandenberghe). This is the same problem as problem 2 in the first assignmen but now you need to solve it in a different way.

A vehicle (say, an airplane) travels along a fixed path of *n* segments, between *n + 1* waypoints labeled *0, . . . , n*. Segment i starts at waypoint *i − 1* and terminates at waypoint *i*. The vehicle starts at time *t = 0* at waypoint *0*. It travels over each segment at a constant (nonnegative) speed; s_i is the speed on segment *i*. We have lower and upper limits on the speeds: $s_{min} \le s \le$

s_{max}. The vehicle does not stop at the waypoints; it simply proceeds to the next segment. The travel distance of segment *i* is d_i (which is positive), so the travel time over segment *i* is d_i/s_i. We let τ_i, $i = 1, \ldots, n$, denote the time at which the vehicle arrives at waypoint *i*. The vehicle is required to arrive at waypoint *i*, for $i = 1, \ldots, n$, between times $\tau_{min,i}$ and $\tau_{max,i}$, which are given. The vehicle consumes fuel over segment *i* at a rate that depends on its speed $\Phi(s_i) = a\, s_i^2 + b\, s_i + c$ kg/s.

You are given the data *d* (segment travel distances), *smin* and *smax* (speed bounds), *τmin* and *τmax* (waypoint arrival time bounds), and the the parameters *a*, *b*, and *c*.
For the given form of the potentials, find the way to reduce the problem to a convex optimization problem and solve it using CVX (**NB:** you need not necessarily use one of the "canonical" convex optimization formulations we saw in the course). Use MATLAB command *stairs* to plot speed vs time for the optimal schedule. What are relative pros and cons for using convex optimization vs. dynamic programming for such task?

Problem 3 (**1 point**) Find the dual problem of the LP in a general form (simplify the dual problem):

$$\min p^T x$$

$$\text{s.t.:} \quad Ax \leq b$$

$$Cx = d$$

Problem 4 (**2 points**). Consider the following convex quadratic program:

$$\min(x_1^2 + x_2^2 - x_1\, x_2)$$

$$\text{s.t. } x_1 + 2x_2 \geq 1$$

$$3x_1 + x_2 \geq 1$$

Derive the dual program. Put both programs into CVX and solve them. Verify that the optimal values are the same. State the KKT conditions. Check that the optimal primal and dual variables satisfy them.

Problem 5. Consider the *binary linear program*:

$$\min c^T x$$

$$\text{s.t. } Ax \leq b$$

$$\forall i\ x_i \in \{0,1\}$$

a) (**1 point**) – using partial dualization, dualize the first set of constraints (Ax <= b). Write the partial dual, and suggest how it can be evaluated efficiently.
b) (**2 points**) – consider the following instance of the problem

$$\min -x_1 - 2x_2 - 4x_3 - 3x_4$$

$$\text{s.t. } 10x_1 + 10x_3 + 10\,x_4 \leq 15$$

$$10x_1 + 10x_2 + 10\,x_4 \leq 13$$

$$x_1, x_2, x_3, x_4 \in \{0,1\}$$

Evaluate the partial dual from part (a) on the grid *mgrid(0:0.05:10,0:0.05:10);* Plot the resulting function using 3d plot (mpl_toolkits.mplot3d, the MATLAB function *surf)*. Rotate the plot and add 2-3 most telling views to the report. Include one of the views that demonstrate whether strong duality holds for your partial dual (explain whether it holds or not). For rotating plot in ipython notebook, turn off matplotlib inline, so that the plot will be in a separate window.