Sharif University of Technology

Department of Electrical Engineering

Convex Optimization I (25756-1)

CHW 1

Instructor: Dr. R. Amiri | Fall Semester 1403 | Max mark: 100

---

1. **DCP Representation with CVX (25 Pts)**

   Disciplined convex programming (DCP) is a system for composing functions while ensuring their convexity. It is the language that underlies CVX. Essentially, each node in the parse tree for a convex expression is tagged with attributes for curvature (convex, concave, affine, constant) and sign (positive, negative) allowing for reasoning about the convexity of entire expressions. Typically, writing problems in the DCP form is trivial, but in some cases manipulation is required to construct expressions that satisfy the rules.

   1. For each set of mathematical expressions below, first briefly explain why each defines a convex set. Then, give an equivalent DCP expression along with a brief explanation of why the DCP expression is equivalent to the original for each set. DCP expressions should be given in a form that passes analysis (a green tick on the left of the expression box) at DCP analyzer. Just in case you are welcome to use visualization.

      (a) $\|(x, y, z)\|_2^2 \leq 1$

      (b) $\sqrt{x^2+1} \leq 3x + y$

      (c) $\frac{1}{x} + \frac{2}{y} \leq 5, x > 0, y > 0$

      (d) $(x + z)y \geq 1, x + z \geq 0, y \geq 0$

      (e) $x\sqrt{y} \geq 1,\ x \geq 0, y \geq 0$

      (f) $\log(e^{y-1} + e^{x/2}) \leq -e^x$

      (g) $\frac{y^2}{x} + z \leq 1$

      (h) $x \log(\frac{x}{y}) \leq x - y,\ \ x, y > 0$

   2. Is the function $f(x) = \log\left(\alpha + \sum_{i=1}^{n} \frac{\beta_i}{x_i}\right) + \lambda\|x\|_2$, where $\alpha \geq 0$, $\beta > 0$, $\lambda > 0$, convex on $\mathbb{R}^n_{++}$? Is it possible to express this into an equivalent DCP expression? Now, consider the following optimization problem:

      $$\begin{array}{ll} \text{minimize} & \log\left(\alpha + \sum_{i=1}^{n} \frac{\beta_i}{x_i}\right) + \lambda\|x\|_2 \\ \text{subject to} & \sum_{i=1}^{n} x_i \leq 1 \end{array}$$

      where $x \in \mathbb{R}^n_{++}$ is the optimization variable, and $\alpha \geq 0$, $\beta > 0$, and $\lambda > 0$ are fixed problem parameters. Can you use CVX to solve the above problem for given α, β, and λ? Explain and Present your documents.

2. **2D Lasso Problem (15 Pts)**

   Using CVX, we will solve the 2d lasso problem and its variants:

   $$\min_{\theta \in \mathbb{R}^{mn}} \frac{1}{2} \sum_{i=1}^{mn} (y_i - \theta_i)^2 + \lambda \sum_{(i,j) \in E} |\theta_i - \theta_j| \,.$$

The set E is the set of all undirected edges connecting horizontally or vertically neighboring pixels in the image. More specifically, $(i, j) \in E$ if and only if pixel i is the immediate neighbor of pixel j on the left, right, above or below.

1. Load the basic test data from `"toy.csv"` and solve the 2d lasso problems with $\lambda = 1$. Report the objective value obtained at the solution and plot the solution and original data as images. Why does the shape change its form? What category of convex problem is it? (LP, QP, SOCP, SDP and so on)
2. Another way to formulate the 2d lasso problem is as follows:

$$\min_{\theta \in \mathbb{R}^{m \times n}} \frac{1}{2} \sum_{a=1}^{m} \sum_{b=1}^{n} (y_{a,b} - \theta_{a,b})^2 + \lambda \sum_{a=1}^{m-1} \sum_{b=1}^{n-1} \left\| \begin{pmatrix} \theta_{a,b} - \theta_{a+1,b} \\ \theta_{a,b} - \theta_{a,b+1} \end{pmatrix} \right\|_p .$$

   Note that the index a, b here refers to the coordinates of pixel i. When taking a 1-norm ($p = 1$), the formulation reduces to the 2d fused lasso mentioned above, and the latter term is called an "anisotropic" total variation penalty. When taking a 2-norm ($p = 2$), the term is called an "isotropic" total variation penalty.

   Solve the "isotropic" 2d lasso problems with $\lambda = 1$ on `"toy.csv"`. Report the objective value obtained at the solution and plot the solution and original data as images. Informally speaking, why is the output different from the "anisotropic" penalty, and what's the difference? What category of convex problem is it? (LP, QP, SOCP, SDP and so on)

   Hint: For `cvxpy` users, the `diff` function, the `hstack` function, and the axis option in the `norm` function would be useful. For Matlab `CVX` users, there is a `norms(x,p,dim)` function that can compute the norm along different dimensions.
3. Next, we consider how the solution changes as we vary λ. Load a grayscale 64×64 pixel image from `"baboon.csv"` and solve the isotropic and anisotropic 2d lasso problem for this image for $\lambda \in \{10^{-\frac{2k+1}{4}} : k = 0, 1, \ldots, 4\}$. For each λ, report the value of the optimal objective value, plot the optimal image and show a histogram of the pixel values (100 bins between values 0 and 1). What change in the histograms can you observe with varying λ for the isotropic and anisotropic penalties?

3. **Group Testing (15 pt)**

In this problem we will explore the idea of *group testing* as a strategy For testing a large population for a rare disease by pooling samples together. Suppose that we have a population of N people and we collect saliva samples from each of them. We are looking for genetic signatures of a particular virus in these samples. Let x_n denote the concentration of this material in the sample for the n^{th} person. We will assume that for healthy people $x_n = 0$, but for infected people, $x_n > 0$. Our testing procedure will be to form a series of M mixtures of samples from different subsets of people, and then only run tests on these mixtures. The goal here is to set $M < N$, and the question is then whether we can identify the infected people from the results of these tests.

We will consider the following approach: we will form mixtures by constructing *random* combinations of samples, and we will attempt to recover the original x using a simple convex optimization problem.

To mathematically represent the sampling/testing process, assume that we will ultimately run M tests, each of which will tell us the concentration of viral material in the combined sample being tested. For each person, their sample will be divided into $K < M$ equal portions, which will be assigned at random to the M tests. We will do this independently for each of the N people. We can ultimately represent the concentration of viral material in each of the mixed samples that we will ultimately test as a vector $\mathbf{y} \in \mathbb{R}^M$. We can write $\mathbf{y}$ as

$$\mathbf{y} = A\mathbf{x},$$

where A is a matrix that represents the assignment of people to mixed samples/tests. Specifically, A is a $M \times N$ matrix where each column is constructed independently by picking K entries at random, setting them to 1, and setting the remaining entries to 0. Suppose there is no noise in our tests, so that we can estimate $\mathbf{y}$ perfectly. Our inference problem is now to estimate $\mathbf{x}$ given knowledge of $\mathbf{y}$ and A. In general, since $M < N$, recovering $\mathbf{x}$ is impossible. However, when $\mathbf{x}$ is sparse, meaning that it has only a few nonzeros (in this case meaning that most of the population is negative), then recovering $\mathbf{x}$ is possible, although this fact was only broadly appreciated within the last 15 years or so.

We will try to estimate x by solving the following optimization problem:

$$\underset{x}{\text{minimize}} \, \|x\|_1 \quad \text{subject to} \quad Ax = y, \quad x \geq 0. \tag{1}$$

Below we will explore when and how well this works.

(a) Suppose that you are testing a population of size $N = 1000$, but you can only process $M = 100$ tests. Assume that only 1% of the population is positive (meaning that there will be 10 infected individuals). Each person's sample will be split and added to $K = 10$ different batches. The file `group_testing.py` contains code that sets up this problem. Use CVXPY to solve the optimization problem above and verify that this approach correctly identifies the 10 infected individuals.

(b) Experiment with K. In practice, you might not want to divide a person's sample into too many tests. How low can you set K before things begin to fail?

(c) Suppose that the prevalence of the disease begins to grow beyond 1%. How widespread can the disease become before the approach begins to fail (holding M and N fixed)? As the disease becomes more widespread, you may need to adjust K. What value of K seems to work best when the spread of the disease is just below the threshold where the approach begins to fail?

4. **Learning Quadratic Metrics from Distance (15 pt)**

Consider a collection of N pairs of points in $\mathbb{R}^n$, represented as $x_1, \ldots, x_N$ and $y_1, \ldots, y_N$. Alongside these, we have a series of positive distances $d_1, \ldots, d_N > 0$.

The objective is to identify a quadratic pseudo-metric d.

$$d(x, y) = \left((x - y)^T P (x - y)\right)^{1/2},$$

with $P \in \mathbf{S}^n_+$, which approximates the given distances, i.e., $d(x_i, y_i) \approx d_i$. (The pseudo-metric d is a metric only when $P \succ 0$; when $P \succeq 0$ is singular, it is a pseudo-metric.)

To do this, we will choose $P \in \mathbf{S}^n_+$ that minimizes the mean squared error objective

$$\frac{1}{N} \sum_{i=1}^{N} (d_i - d(x_i, y_i))^2.$$

(a) Explain how to determine P using convex or quasiconvex optimization methods. If you are unable to provide an exact formulation (one that guarantees minimization of the total squared error objective), propose an approximate formulation that minimizes the objective function as closely as possible, while adhering to the given constraints.

(b) Calculate the optimal matrix P using the training data provided in the `LQM_data.npz` file, and then output the mean squared error on the test data. You can load the `LQM_data.npz` file using `numpy.load`.

5. **Fitting a sphere to data (15 pt)**
Consider the problem of fitting a sphere $\{x \in \mathbf{R}^n \mid \|x - x_c\|_2 = r\}$ to m points $u_1, \ldots, u_m \in \mathbf{R}^n$, by minimizing the error function

$$\sum_{i=1}^{m} (\|u_i - x_c\|_2^2 - r^2)^2$$

over the variables $x_c \in \mathbf{R}^n$, $r \in \mathbf{R}$.

(a) Explain how to solve this problem using convex or quasiconvex optimization. The simpler your formulation, the better. (For example: a convex formulation is simpler than a quasiconvex formulation; an LP is simpler than an SOCP, which is simpler than an SDP.) Be sure to explain what your variables are, and how your formulation minimizes the error function above.

(b) Use your method to solve the problem instance with data given in the file `Q5.npz`, with $n = 2$. Plot the fitted circle and the data points.

6. **Minimax Rational Fit to the Exponential (15 pt)**
We consider the problem with data

$$t_i = -3 + 6(i-1)/(k-1), \quad y_i = e^{t_i}, \quad i = 1, \ldots, k,$$

where $k = 201$. (In other words, the data are obtained by uniformly sampling the exponential function over the interval $[-3, 3]$.) Find a function of the form

$$f(t) = \frac{a_0 + a_1 t + a_2 t^2}{1 + b_1 t + b_2 t^2}$$

that minimizes $\max_{i=1,\ldots,k} |f(t_i) - y_i|$. (We require that $1 + b_1 t_i + b_2 t_i^2 > 0$ for $i = 1, \ldots, k$.)
Find optimal values of a_0, a_1, a_2, b_1, b_2, and give the optimal objective value, computed to an accuracy of 0.001.