ECE 3803, Fall 2021

Homework #7

Due Sunday, December 5, at 11:59pm

1. Prepare a one paragraph summary of what we talked about in class since the last assignment. I do not want just a bulleted list of topics, I want you to use complete sentences and establish context (Why is what we have learned relevant? How does it connect with other classes?). The more insight you give, the better.

2. In the problems below you will make use of the CVXPY package. You will probably need to install this package as it does not come standard with most python distributions. This should be easy – see https://cvxpy.readthedocs.io/en/latest/install/index.html for OS-specific instructions. Install CVXPY and poke around on the CVXPY website to get a feeling for what it can do.

3. 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:[1] we will form mixtures by constructing *random* combinations of samples, and we will attempt to recover the original $\boldsymbol{x}$ 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 $\boldsymbol{y} \in \mathbb{R}^M$. We can write $\boldsymbol{y}$ as

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

   where $\boldsymbol{A}$ is a matrix that represents the assignment of people to mixed samples/tests. Specifically, $\boldsymbol{A}$ 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 $\boldsymbol{y}$ perfectly. Our inference problem is now to estimate $\boldsymbol{x}$ given knowledge of $\boldsymbol{y}$ and $\boldsymbol{A}$. In general, since $M < N$,

[1] Just to be clear, this is not exactly what Georgia Tech is doing, but it is similar in spirit. The approach that Georgia Tech takes involves a more careful design of the "mixing matrix" $\boldsymbol{A}$ that enables a simpler strategy to identify who is infected. This approach actually builds on ideas that date back to World War II, when they were used to test soldiers for syphilis.

recovering $\boldsymbol{x}$ is impossible. However, when $\boldsymbol{x}$ is sparse, meaning that it has only a few nonzeros (in this case meaning that most of the population is negative), then recovering $\boldsymbol{x}$ *is* possible, although this fact was only broadly appreciated within the last 15 years or so.

We will try to estimate $\boldsymbol{x}$ by solving the following optimization problem:

$$\underset{\boldsymbol{x}}{\text{minimize}} \ \|\boldsymbol{x}\|_1 \qquad \text{subject to} \qquad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}, \quad \boldsymbol{x} \geq \boldsymbol{0}.$$

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 `hw07_prob03.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, N, and K fixed and choosing K to be something reasonable based on part (b)).

4. Here we explore Lagrangian duality with a simple example. Consider the optimization problem

$$\begin{aligned} &\underset{x \in \mathbb{R}}{\text{minimize}} \ x^2 + 1 \\ &\text{subject to} \ (x-2)(x-4) \leq 0. \end{aligned}$$

(a) Provide as simple as possible of a description of the feasible set.

(b) Determine both the minimizer $x^\star$ as well as the value of the objective function at the minimizer.

(c) Plot the objective function, indicating in your plot the feasible set. Also plot the Lagrangian $\mathcal{L}(x, \lambda)$ for a few values of λ.

(d) Derive and plot the dual function $d(\lambda)$.

(e) State the dual problem and find the maximizer $\lambda^\star$ as well as the value $d(\lambda^\star)$. Does strong duality hold?

5. In this problem we will explore two alternative approaches to solving a simple variant of the least squares problem where we add the constraint that the solution is non-negative, i.e., we wish to solve

$$\underset{\boldsymbol{x} \in \mathbb{R}^N}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 \qquad \text{subject to} \qquad \boldsymbol{x} \geq \boldsymbol{0}.$$

This is natural in many practical applications where the entries of $\boldsymbol{x}$ have physical interpretations (e.g., light intensity, power, concentration of some physical material, etc.) that don't really make sense as negative quantities. Below we will assume throughout that $\boldsymbol{A}$ is an $M \times N$ matrix with $M > N$ and that $\boldsymbol{A}$ is full rank.

(a) Derive the Lagrangian function for this optimization problem (pay careful attention to the sign of each term).

(b) Show that the dual optimization problem is itself another nonnegative least squares problem. Specifically, show that the dual problem can be expressed as

$$\underset{\boldsymbol{\lambda} \in \mathbb{R}^N}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{y}' - \boldsymbol{A}'\boldsymbol{\lambda}\|_2^2 \qquad \text{subject to} \qquad \boldsymbol{\lambda} \geq \mathbf{0},$$

where $\boldsymbol{y}' = (\boldsymbol{A}^T\boldsymbol{A})^{-\frac{1}{2}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}$ and $\boldsymbol{A}' = -(\boldsymbol{A}^T\boldsymbol{A})^{-\frac{1}{2}}$. Do this by deriving the dual function $d(\boldsymbol{\lambda})$ by first finding the $\boldsymbol{x}$ that minimizes the Lagrangian $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$, and then plugging this into $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$.

(c) Recall from the notes that one of the KKT conditions (KKT4) is that if $\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star$ are primal/dual optimal, then $\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star) = \mathbf{0}$. Write out a simple expression of this condition for this problem.

(d) Write out a simple expression for the "complementary slackness" KKT condition (KKT3) for this problem.

(e) Try solving a nonnegative least squares problem using CVXPY. The file `hw07_prob05.py` contains some code to set up a nonnegative least squares problem and then solve it using CVXPY. Make sure you understand what the code is doing, and verify that the resulting solution satisfies the KKT conditions from parts (c) and (d).

(f) Implement the projected gradient descent approach described on page 23 of the last block of notes. This should be a minor variation on something you have already implemented before, but there are two important caveats. First, when we solved least squares problems before using gradient descent, we calculated the optimal step size α_k at each iteration. This is much tougher to do in this case because the actual optimal step size would involve figuring out what α is optimal after accounting for the projection step, and this is not easy to do in closed form. You should either use a line search to choose α, or just take a fixed step size. For guidance, the theory guarantees convergence if $\alpha \leq 1/\|\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\|_2$.

The second challenge here involves defining a stopping criterion. You cannot expect that the norm of the gradient will be zero at the solution. Instead you could either define a stopping criterion involving $\|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\|_2$, or alternatively you could do something inspired by part (c) above.

Verify that the resulting solution satisfies the KKT conditions from parts (c) and (d).