CS 791 Project

General Guidelines:

- For each problem, please provide, at the end of your report, a commented version of your Matlab files. Please also prepare a **typed** report that describes what you did. The report should be as concise as possible while providing all necessary information required to replicate your plots. The plots should contain multiple curves and can be formatted so that many plots can fit on one page (so that your report is not longer than it should be). Provide the expressions for the gradients and Hessians used in your algorithms, in your report. Also be clear about what the algorithms are (you can choose to express them in pseudocode). **The project report will be due May 14, 2018**. But as you will notice below, there is a lot to be done, and therefore you should start early.

Part I: Basics of convex optimization

1. *Gradient Descent Method*: Consider the following optimization problem:

$$\text{minimize } \frac{1}{2}x^T Q x + q^T x$$

where Q is an $n \times n$ positive definite matrix, and it is not necessarily a diagonal matrix. Note that the solution to this problem is $x^* = -Q^{-1}q$. As an alternative, design a gradient descent algorithm that will solve this unconstrained problem. Please pay attention to the following:

(a) Your algorithm should take Q and q as input and return x^*.

(b) Generate figures that look like Fig 9.6 in the textbook with different starting points.

(c) You should implement both exact line search and backtracking line search and compare the two. Also, when using backtracking line search try different α and β and discuss their effect on the convergence.

(d) Try different problem sizes n and comment on its effect on the convergence.

(e) Investigate the effect of the condition number of Q on the convergence behavior of your algorithm. As the condition number increases, does the convergence become faster, or slower? Give examples to illustrate this effect.

(f) Design a steepest descent algorithm with the choice of norm is given as given at the bottom of pp. 476 of our textbook, where P is selected as a diagonal matrix, whose diagonal elements are the same as those of Q. Is this new algorithm as sensitive to the condition number of Q?

(g) Comment on how the gradient descent algorithm can be used to solve a least-squares problem and give an example of a least-square problem that you solve with this algorithm. Also, comment on the computational advantages of using the gradient descent algorithm as compared with just solving the linear system of equations $x^* = -Q^{-1}q$. Do these advantages remain when the steepest descent approach described above is used? What are some potential disadvantages to using gradient descent, or steepest descent as compared with just solving the linear system of equations?

2. *Newton's Method for Unconstrained Problems*: Replicate Fig. 9.20 for the objective function in equation (9.20) in the textbook. Please include the expressions for the gradient and the Hessian in your report. Also add a few other initial starting points.

3. *Newton's Method with Equality Constraints*: Consider the equality constrained entropy maximization problem

$$\begin{array}{ll} \text{minimize} & f(x) = \sum_{i=1}^{n} x_i \log x_i \\ \text{subject to} & Ax = b \end{array}$$

with $\mathbf{dom} f = \mathbf{R}^n_{++}$ and $A \in \mathbf{R}^{p \times n}$, with $p < n$. Generate a problem instance with $n = 100$ and $p = 30$ by choosing A randomly (checking that it has full rank), choosing $\hat{x}$ as a random positive vector (e.g., with entries uniformly distributed on [0, 1]) and then setting $b = A\hat{x}$. (Thus, $\hat{x}$ is feasible.)

Compute the solution of the problem using *Infeasible start Newton method*. You can use initial point $x^{(0)} = \hat{x}$ (to compare with the *standard Newton method*), and also the initial point $x^{(0)} = \mathbf{1}$.

Note: You need to implement the elimination method to solve KKT systems.

4. *Feasibility of an LP*: Develop an algorithm to test the feasibility of an LP using the approach in (11.19) where $f_i(x)$ are affine functions of x. Express the gradients and the Hessians needed for the algorithm explicitly in your report and describe the algorithm.

Note: Solve Example 11.4 and generate Figure 11.9 by the basic phase I algorithm. Examine the choice of μ (see Figure 11.4) in the simulation.

5. A cvx *Experiment in Compressive Sensing*: Consider the following problem

$$\begin{array}{ll} \text{minimize} & \|x\|_1 \\ \text{subject to} & y = \Phi x \end{array}$$

where Φ is a $k \times n$ matrix with $k \ll n$, and x is a vector with only S nonzero elements. The interpretation is that y constitutes our k measurements of a sparse vector x, through a "measurement matrix" Φ. Since Φ is fat, there are infinitely many vectors x that satisfy the equality constraint in the problem, so we cannot determine x uniquely, only from the constraint. But if we know that x is sparse (i.e., only S nonzero elements out of n), and if k is large enough (i.e., we have enough 'projections' of x), then the optimization problem above can uniquely determine x. Note that we do not need to know the sparsity pattern (which elements of x are nonzero).

For the purposes of this problem, we will assume that Φ consists of the rows of a DFT matrix which has an element in the l^{th} row and m^{th} column given by $[\Phi]_{l,m} = n^{-1/2}\exp(-j2\pi f_l m/N)$, where $f_l \in \{0,1,\dots,n-1\}$ for $l = 1,\dots,k$. The interpretation is that x is being "sensed" or "sampled" in k different frequencies. Recall that $k < n$, so we have much fewer frequency samples than the length of x, hence the name compressed sampling or compressed sensing. An interesting result in connection with the problem above is that if k frequencies are chosen randomly and uniformly in the set $\{0,1,\dots,n-1\}$, and the number of frequency samples are at least

$$k > CS \log n,$$

then the above optimization problem has a solution which will reconstruct x perfectly, with high probability. The idea is that for almost all selection of k frequencies out of the n possible, perfect reconstruction of the sparse x is possible with much fewer than n, but more than $CS \log n$ samples, where C is just some constant,

independent of n and k. In short, if k is sufficiently large, a random selection of frequencies will with very high probability yield a sampling matrix Φ that will generate a y which can be used to recover x perfectly using the optimization problem above. This probability will approach 1 very rapidly especially if n is large as well. However, even when $k > CS \log n$ is satisfied, it is possible to select the frequencies such that we cannot reconstruct x (even though this occurs with ever smaller probability when n is large).

(a) Give an example of a set of k frequencies for which we would only be measuring a y vector that would all be zeros, even though k could be as high as $n - S$ (which is clearly $> CS \log n$). To find such an example, think of a comb-shaped signal in discrete-time.

(b) Select n, S, and generate Φ randomly using the description above, where the k frequencies are selected randomly, uniformly distributed in $\{0,1, \dots, n-1\}$. Use cvx to recover x. For values of $n = 50$, and $n = 100$, determine the smallest value of k that will recover x perfectly from randomly selected frequencies.

(c) Suppose we now have noisy measurements so that $y = \Phi x + n$, where n is an unknown noise vector that satisfies $\|n\|_2 \leq \epsilon$. Use a Φ matrix and k value and set of frequencies that enable perfect recovery in the noiseless case to generate your data. For this data solve the following problem in cvx:

$$\begin{array}{ll} \text{minimize} & \|x\|_1 \\ \text{subject to} & \|y - \Phi x\|_2 \leq \epsilon \end{array}$$

Plot $\|x^* - x\|_2$ versus ϵ for a reasonable range of ϵ, keeping Φ, x the same.

Part II: Application of convex optimization on your research

Please find a research problem in your area that requires optimization techniques to solve the problem. For this problem, you need to explain the problem and formulate it as an optimization problem. Then you need to discuss how this problem is solved.

Note: The problem does not have to be convex and you may use other techniques beyond this course.