EEE 598 – Project

General Guidelines:

Please do all of the following problems provided below. For each case, 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 chose to express them in pseudocode). The project report will be due the final exam time. But as you will notice below, there is a lot to be done, and therefore you should start early. You should show a serious effort on **all problems**.

The project consists of 5 parts as enumerated below. The last 4 parts involve writing Matlab code for solving convex optimization problems using algorithms we learned in class. The first part involves using an already existing convex optimization solver (`cvx`) to study a simple example of Compressive Sensing. To learn how to install `cvx` on top of Matlab, please go to `http://www.stanford.edu/~ boyd/cvx/`. You can also use `cvx` to check your answers for parts 2-5, even though it is not mandatory.

1. *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 a matrix with random entries chosen as independent, plus or minus one with equal probability. Recall that $k < n$, so we have much fewer "samples" (inner products with Bernoulli vectors) than the length of x, hence the name compressed sampling or compressed sensing. The idea is that for almost all selection of k samples, 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. Loosely speaking, if k is sufficiently large, the Bernoulli matrix 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. Toward this goal, select n, S, and generate Φ randomly using the description above, Use `cvx` to recover x. For values of $n = 50$, $n = 100$, and $n = 500$, and values of S (small) that you chose, plot the probability of perfect recovery versus k. To estimate this probability, you need to run this experiment many times and count how many times you get perfect reconstruction. In other words, you need to use a Monte Carlo approach to estimate this probability.

2. *Least Squares with Gradient Descent:* Consider a least-squares problem with a matrix A and vector b. Formulate this as a quadratic minimization problem and use a gradient descent algorithm to solve the least-squares problem. Your gradient descent algorithm should not involve any matrix inversions. Do an example where A is 3×2, and show figures that look like Figures 9.5 and 9.6 on pp. 472-473.

3. *Newton's Method for Unconstrained Problems:* Use Newton's algorithm with backtracking line search to minimize the function $f(x_1, x_2) = \exp(x_1 + 2x_2) + \exp(x_1 - 2x_2) + \exp(-x_1)$. Use backtracking line search. Show a plot of the function value minus $p^\star$ versus the number of iterations.

4. *Newton's Method with Equality Constraints:* Implement the infeasible start Newton method for $f(x_1, x_2)$ above with the constraint $x_1 = x_2$.

5. *Separating Two Sets of Points via Feasibility of an LP:* Consider two sets of points $\{x_1, \ldots, x_N\}$ and $\{y_1, \ldots, y_M\}$, where each point is two dimensional (so we have $N + M$ points on the plane). We say that these sets are linearly separable if there is a line such that all the xs are on one side of the line, and all the ys are on the other side. Express this as a linear feasibility problem (see eqn 8.21). Develop an algorithm *from scratch* to solve this problem 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. Show that for two simple examples with two variables (one feasible, and one infeasible) that your algorithm can identify feasibility.