Homework #5 (Programming)

Due April 21 (Friday) midnight 11:59pm

In this homework, you will implement several algorithms in Matlab or Python whichever you prefer. You should turn in a zip folder (IE521_HW5_lastname_firstname.zip) to the TA, which should contain all output figures and source codes.

Problem 1: Logistic Regression

Logistic regression is another popular model in machine learning used for classification. Mathematically, given a training dataset of \( m \) points \((x_1, y_1), \ldots, (x_m, y_m)\) where \( x_i \in \mathbb{R}^n \) stands for the feature vector and \( y_i \in \{1, -1\} \) stands for two classes, the model aims to solve the following optimization problem:

\[
\min_w f(w) := \sum_{i=1}^{m} \log(1 + \exp(-y_i w^T x_i)) + \frac{\lambda}{2} \|w\|^2 \tag{P}
\]

where the parameter \( \lambda > 0 \). Note that this problem is indeed an unconstrained convex minimization problem with a twice-differentiable and strongly convex objective.

Exercise 1.1 (Gradient Descent)  
Implement the simplest Gradient Descent:

\[
w_{k+1} = w_k - \gamma_k \nabla f(w_k)
\]

where \( \gamma_k > 0 \) is the stepsize. Use fixed stepsize \( \gamma_k = \gamma \leq 1/L \), where \( L \) is such that \( \nabla^2 f(w) \leq L \cdot I \). In fact, we can verify that \( L \leq \frac{1}{4} \lambda_{\text{max}}(X^T X) + \lambda \) for the objective in (P). Terminate the algorithm when the criterion \( \|\nabla f(x)\|_2 \leq \epsilon \) is met. Your implementation should produce a function that can be called as

\[
[w_{\text{opt}}, f_{\text{hist}}] = \text{gradient\_descent}(X, y, \lambda, w_0, \gamma, \epsilon)
\]

where
- \( w_{\text{opt}} \) is the output approximate solution
- \( f_{\text{hist}} \) is the history of function values \( \{f(x_k), k = 0, 1, 2, \ldots\} \).
- \( X \) is the input data matrix \((m \times n)\), \( y \) is the label vector \((m \times 1)\), \( \lambda \) is the regularization parameter
- \( w_0 \) is the initial point, \( \gamma \) is the fixed stepsize, \( \epsilon \) is the targeted accuracy

Exercise 1.2 (Newton Method)  
Implement the basic Newton method:

\[
w_{k+1} = w_k - [\nabla^2 f(w_k)]^{-1} \nabla f(w_k)
\]

Terminate the algorithm when the criterion \( \|\nabla f(x)\|_2 \leq \epsilon \) is met. Your implementation should produce a function that can be called as

\[
[w_{\text{opt}}, f_{\text{hist}}] = \text{Newton}(X, y, \lambda, w_0, \epsilon)
\]

Exercise 1.3 (Test on Real Dataset)  
Apply the two algorithms on WDBC dataset provided in HW 3. Set \( \lambda = 1 \), \( \epsilon = 10^{-3} \).

- Try different stepsize in Gradient Descent, and observe the behavior of the algorithm.
- Try different random initialization in Newton Method, and observe the behavior of the algorithm.

Now initialize both algorithms with same starting point, and generate a plot that compares the trajectory of objective values from both algorithms.
Problem 2: Linear Program

Suppose we want to solve the linear program

$$\min \{c^T x : Ax \leq b\}.$$  

Exercise 2.1 (Damped Newton Method) Consider the unconstrained self-concordant minimization problem that approximates the linear program

$$\min_x f(x) := c^T x - \sum_{i=1}^{m} \log(b_i - a_i^T x) \quad (SCP)$$

Implement the damped Newton method to solve (SCP):

$$w_{k+1} = w_k - \frac{1}{1 + \lambda_k} [\nabla^2 f(w_k)]^{-1} \nabla f(w_k)$$

where $$\lambda_k = \sqrt{\nabla f(w_k)^T [\nabla^2 f(w_k)]^{-1} \nabla f(w_k)}$$. Terminate the algorithm when Newton decrement $$\lambda_f(x) \leq \epsilon$$. Your implementation should produce a function that can be called as

$$[x_{\text{opt}}, f_{\text{hist}}] = \text{damped\_Newton}(A, b, c, x_0, \epsilon)$$

Now generate a problem instance by choosing $$A$$, $$b$$, $$c$$ randomly with $$m = 100$$, $$n = 50$$, say

$$a_{ij} \sim \mathcal{N}(0, 1), b_i \sim \text{Uniform}(0, 1), c_{ij} \sim \mathcal{N}(0, 1), i = 1, \ldots, m, j = 1, \ldots, n$$

- First run the algorithm with initial point $$x_0 = 0$$ and accuracy $$\epsilon = 10^{-10}$$, and get an approximate value for $$f^*$$.  
- Then run the algorithm with initial point $$x_0 = 0$$ and accuracy $$\epsilon = 10^{-4}$$. Generate a plot that shows the approximate function gap $$f(x_k) - f^*$$ versus iteration, and another plot that shows the Newton decrement $$\lambda_k$$ versus iteration.