Instructions: (Please read carefully and follow them!)

Try to solve all problems on your own. If you have difficulties, ask the instructor or TAs.

In this session, we will apply the methods we have developed in the previous labs, to solve a practical problem. The scalability analysis performed in previous labs will be carried out in this lab as well.

The implementation of the optimization algorithms in this lab will involve extensive use of the `numpy` Python package. It would be useful for you to get to know some of the functionalities of `numpy` package. For details on `numpy` Python package, please consult https://numpy.org/doc/stable/index.html

For plotting purposes, please use `matplotlib.pyplot` package. You can find examples in the site https://matplotlib.org/examples/.

Please follow the instructions given below to prepare your solution notebooks:

- Please use different notebooks for solving different Exercise problems.
- The notebook name for Exercise 1 should be `YOURROLLNUMBER_IE684_Lab6_Ex1.ipynb`.
- Similarly, the notebook name for Exercise 2 should be `YOURROLLNUMBER_IE684_Lab6_Ex2.ipynb`, etc.
- Please post your doubts in MS Teams Discussion Forum channel so that TAs can clarify.

There are only 2 exercises in this lab. Try to solve all problems on your own. If you have difficulties, ask the Instructors or TAs.

Only the questions marked **[R]** need to be answered in the notebook. You can either print the answers using `print` command in your code or you can write the text in a separate text tab. To add text in your notebook, click `+Text`. Some questions require you to provide proper explanations; for such questions, write proper explanations in a text tab. Some questions require the answers to be written in LaTeX notation. Please see the demo video (posted in Lab 1) to know how to write LaTeX in Google notebooks. Some questions require plotting certain graphs. Please make sure that the plots are present in the submitted notebooks. Please include all answers in your `.pynb` files.

After completing this lab's Exercise 1, click File → Download .ipynb and save your files to your local laptop/desktop. Create a folder with name `YOURROLLNUMBER_IE684_Lab6_Ex1` and copy your `.ipynb` files to the folder. Then zip the folder to create `YOURROLLNUMBER_IE684_Lab6_Ex1.zip`. Then upload only the `.zip` file to Moodle.

Similarly, after completing this lab's Exercise 2, click File → Download .ipynb and save your files to your local laptop/desktop. Create a folder with name `YOURROLLNUMBER_IE684_Lab6_Ex2` and copy your `.ipynb` files to the folder. Then zip the folder to create `YOURROLLNUMBER_IE684_Lab6_Ex2.zip`. Then upload only the `.zip` file to Moodle.

There will be extra marks for students who follow the proper naming conventions in their submissions.

Please check the **submission deadline announced in moodle.**

Exercise 0: Convergence of sequences: (ONLY FOR PRACTICE, NOT FOR SUBMISSION)

Let $\{x^k\}$ be a sequence in $\mathbb{R}^n$ that converges to x^*. One possible classification of the convergence behavior is stated below:

- The convergence is *Q-Linear* if there is a constant $r \in (0, 1)$ such that

$$\frac{||x^{k+1} - x^*||}{||x^k - x^*||} \leq r, \text{ for all } k \text{ sufficiently large,}$$

  where the norm $|| \cdot ||$ is generally assumed to be the Euclidean norm.

- The convergence is *Q-superlinear* if

$$\lim_{k \to \infty} \frac{||x^{k+1} - x^*||}{||x^k - x^*||} = 0.$$

- The convergence is *Q-quadratic* if there exists some $M > 0$ such that we have

$$\frac{||x^{k+1} - x^*||}{||x^k - x^*||^2} \leq M, \text{ for all } k \text{ sufficiently large.}$$

1. For each of following three sequences check empirically how fast they converge and try to check the theoretical rates of convergence:

   (a) $(1 + 0.005)^{-(2^k)}, \; k = 0, 1, 2, \ldots$

   (b) $1 + (0.005)^k, \; k = 0, 1, 2, \ldots$

   (c) $1 + (0.005)^{(-k)}, \; k = 0, 1, 2, \ldots$

   (d) $1 + k^{-k}, \; k = 0, 1, 2, \ldots$ (assume $0^0 = 1$)

   You can check the rates by plotting the iterates.

In this lab, we will consider the ordinary least squares regression (OLSR) problem. We will discuss a few optimization algorithms to solve it. Please use **only** Python as your programming language.

Preparation Exercise (PREP):

1. Import the required Python packages using the following commands
   `import numpy as np`
   `import matplotlib.pyplot as plt`
   (Recall the functionality of these packages.)
2. For replication purposes, initialize the random number generator using `np.random.seed(1000)`
3. Use the `np.random.randn` function to create A as a random numpy array of 1000 rows and 10 columns (recall the purpose of `np.random.randn` function). In data science parlance, we shall call A to be a data set of 1000 data points, each of dimension 10.
4. Create $\bar{x} = \begin{bmatrix}\bar{x}_1 \ \bar{x}_2 \ \dots \ \bar{x}_{10}\end{bmatrix}^\top$ as a random vector of size 10×1 such that when i is odd, each component $\bar{x}_i$ of $\bar{x}$ is sampled uniformly from $[-(i+1), -i] \subset \mathbb{R}$, and when i is even, each component $\bar{x}_i$ of $\bar{x}$ is sampled uniformly from $[i, i+1] \subset \mathbb{R}$ (Use appropriate `numpy` function here).
5. Use `np.random.randn` to create ε as a random vector of size 1000×1.
6. Compute $y = A\bar{x} + \varepsilon$. Use an appropriate numpy function to do the matrix multiplication $A\bar{x}$ efficiently.

Exercise 1: Direct least squares loss minimization

Note that y is a noisy version of $A\bar{x}$. We will now try to estimate $\bar{x}$ assuming that we are given y and A. One possible approach is to solve the following problem:

$$\min_x f(x) = \frac{1}{2}\|Ax - y\|_2^2. \tag{1}$$

The loss term $\|Ax - y\|_2^2$ is called the ordinary least squares (OLS) loss and the problem (1) is called the OLS Regression problem.

1. Write Python functions using appropriate `numpy` routines to compute the objective function value, the gradient value and the Hessian of f.

2. **[R]** With a starting point of $x^0 = \begin{bmatrix} 0 & 0 & \dots 0 \end{bmatrix}^\top \in \mathbb{R}^{10}$, solve problem (1) using the Newton's method implemented with backtracking line search (use $\alpha^0 = 0.99, \rho = 0.5, \gamma = 0.5$ for backtracking line search and $\tau = 10^{-5}$). Comment on difficulties (if any) you face when computing the inverse of Hessian (recall that you need to use an appropriate Python function to compute the inverse of the Hessian). If you face difficulty in computing inverse of Hessian, try to think of some remedy so that you can avoid the issue.

   - Let x^* be the final optimal solution provided by your algorithm. Report the values of x^* and $\bar{x}$, and discuss the observations.
   - Plot the values $\log(||x^k - x^*||_2)$ against iterations $k = 0, 1, 2, \dots$.
   - Prepare a different plot for plotting $\log(|f(x^k) - f(x^*)|)$ obtained from Newton's method against the iterations.
   - Comment on the convergence rates of the iterates and the objective function values, by recalling the definitions given above.

3. **[R]** With a starting point of $x^0 = \begin{bmatrix} 0 & 0 & \dots 0 \end{bmatrix}^\top \in \mathbb{R}^{10}$, solve problem (1) using the BFGS method implemented in the previous lab with backtracking line search (use $\alpha^0 = 0.99, \rho = 0.5, \gamma = 0.5$ for backtracking line search and $\tau = 10^{-5}$).

   - Let x^* be the final optimal solution provided by your BFGS algorithm. Report the values of x^* and $\bar{x}$, and discuss the observations.
   - Plot the values $\log(||x^k - x^*||_2)$ against iterations $k = 0, 1, 2, \dots$.
   - Prepare a different plot for plotting $\log(|f(x^k) - f(x^*)|)$ obtained from BFGS method against the iterations.
   - Comment on the convergence rates of the iterates and the objective function values obtained by BFGS method.

4. **[R]** Compare and contrast the results obtained by Newton's method and BFGS method and comment on the time taken by both the methods.

Exercise 2: Regularized least squares loss minimization

1. Let us now introduce the following regularized problem (with $\lambda > 0$):

$$\min_x f_\lambda(x) = \frac{\lambda}{2} x^\top x + \frac{1}{2}\|Ax - y\|_2^2. \tag{2}$$

   [R] Comment on the significance of the newly added regularizer term $\frac{\lambda}{2}x^\top x$, when compared to problem (1).

2. Write Python functions to compute the function value, gradient and Hessian of f_λ.

3. For $\lambda \in \{10^{-3}, 10^{-2}, 10^{-1}, 1\}$, perform the following: with a starting point of $x^0 = \begin{bmatrix}0 \ 0 \ \dots 0\end{bmatrix}^\top \in \mathbb{R}^{10}$, solve the problem (2) using Newton and BFGS methods with backtracking line search (use $\alpha^0 = 0.99, \rho = 0.5, \gamma = 0.5$ for backtracking line search and $\tau = 10^{-5}$).

4. **[R]** For Newton's method prepare the following plots and discuss the relevant observations:
   (a) Prepare a single plot where you depict the values $\log(||x^k - x^*||_2)$ against iterations $k = 0, 1, 2, \dots$, for each value of λ (use different colors for different λ values; if necessary, add zoomed versions of the plots to depict the behavior clearly, and use appropriate legend in your plots). Comment on the convergence rates of the iterates for each value of λ.
   (b) Prepare a different plot for plotting $\log(|f(x^k) - f(x^*)|)$ against the iterations, for each value of λ (use different colors for different λ value; if necessary, add zoomed versions of the plots to depict the behavior clearly, and use appropriate legend in your plots). Comment on the convergence rates of the objective function values.

5. **[R]** For BFGS method prepare the following plots and discuss the relevant observations:
   (a) Prepare a single plot where you depict the values $\log(||x^k - x^*||_2)$ against iterations $k = 0, 1, 2, \dots$, for each value of λ (use different colors for different λ values; if necessary, add zoomed versions of the plots to depict the behavior clearly, and use appropriate legend in your plots). Comment on the convergence rates of the iterates for each value of λ.
   (b) Prepare a different plot for plotting $\log(|f(x^k) - f(x^*)|)$ against the iterations, for each value of λ (use different colors for different λ value; if necessary, add zoomed versions of the plots to depict the behavior clearly, and use appropriate legend in your plots). Comment on the convergence rates of the objective function values.

6. **[R]** Compare and contrast the results obtained by Newton's method and BFGS method and comment on the time taken by both the methods for each value of λ.