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 continue with the implementation of gradient descent algorithm to solve problems of the form $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$. Recall that gradient descent takes a large number of iterations for some problems. In this lab, we will see some techniques to make gradient descent converge to the optimal point faster.

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

In some cases you might need to use the matrix square root function `sqrtm` from the package `scipy.linalg`. You can use `from scipy.linalg import sqrtm` and then use `sqrtm(A)` to find the matrix square root of matrix `A`.

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_Lab3_Ex1.ipynb`.
- Similarly, the notebook name for Exercise 2 should be `YOURROLLNUMBER_IE684_Lab3_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 exercises, click File → Download .ipynb and save your files to your local laptop/desktop. Create a folder with name `YOURROLLNUMBER_IE684_Lab3` and copy your `.ipynb` files to the folder. Then zip the folder to create `YOURROLLNUMBER_IE684_Lab3.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 1: Gradient Descent with Scaling

Recall that in the last lab, when we tried to solve certain problems of the form $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$ using gradient descent algorithm, we noticed that the algorithm needed a large number of iterations to find the minimizer. As a workaround to avoid such situations, we consider the modified gradient descent scheme illustrated in Algorithm 1.

Input: Starting point x^0, Stopping tolerance τ
Initialize $k = 0$
$\mathbf{p}^k = -\nabla f(\mathbf{x}^k)$.
while $\|\mathbf{p}^k\|_2 > \tau$ **do**
  Choose a suitable scaling matrix $\mathbf{D}^k$.
  $\eta^k = \arg\min_{\eta\geq 0} f(\mathbf{x}^k + \eta\mathbf{D}^k\mathbf{p}^k) = \arg\min_{\eta\geq 0} f(\mathbf{x}^k - \eta\mathbf{D}^k\nabla f(\mathbf{x}^k))$
  $\mathbf{x}^{k+1} = \mathbf{x}^k + \eta^k\mathbf{D}^k\mathbf{p}^k = \mathbf{x}^k - \eta^k\nabla\mathbf{D}^k\nabla f(\mathbf{x}^k)$
  $k = k + 1$
end
Output: $\mathbf{x}^k$.

Algorithm 1: Gradient Descent Procedure with Scaling

---

1. Note that in the notebook file shared with you, we provide the motivation for the modified gradient descent procedure with scaling.
2. [**R**] Consider the function $f(\mathbf{x}) = 1000x_1^2 + 40x_1x_2 + x_2^2$. Write code to find the Hessian matrix of f and its condition number.
3. Note that the update step in the modified gradient descent scheme uses a scaled gradient. Thus it becomes important to set up some criteria for choosing the $\mathbf{D}^k$ matrix in every iteration. In this exercise, we will assume $\mathbf{D}^k$ to be a diagonal matrix. The following questions will help in designing a suitable $\mathbf{D}^k$.
4. [**R**] Based on our discussion on condition number and the derivation of the gradient descent scheme with scaling, can you identify and write down the matrix $\mathbf{Q}$ whose condition number needs to be analyzed in the new gradient scheme with scaling?
5. [**R**] Based on the matrix $\mathbf{Q}$, can you come up with a useful choice for $\mathbf{D}^k$ (assuming $\mathbf{D}^k$ to be diagonal)?
6. Write all related code to implement Algorithm 1 to find the minimizer of the function $f(\mathbf{x}) = 1000x_1^2 + 40x_1x_2 + x_2^2$.
7. [**R**] Note down the minimizer and minimum function value of $f(\mathbf{x}) = 1000x_1^2 + 40x_1x_2 + x_2^2$.
8. [**R**] With starting point $\mathbf{x}^0 = (1, 2000)$ and a stopping tolerance $\tau = 10^{-9}$, find the number of iterations taken by the gradient descent algorithm (without scaling) with exact line search, gradient descent algorithm (without scaling) with backtracking line search, gradient descent algorithm (with scaling) with backtracking line search. For backtracking line search, use $\alpha^0 = 1, \rho = 0.5, \gamma = 0.5$. Note the minimizer and minimum objective function value in each case. Comment on your observations.
9. [**R**] With starting point $\mathbf{x}^0 = (1, 2000)$ and $\tau = 10^{-9}$, we will now study the behavior of gradient descent algorithm (without scaling) with backtracking line search, gradient descent algorithm (with scaling) with backtracking line search, for different choices of α^0. Take $\gamma = \rho = 0.5$. Try $\alpha^0 \in \{1, 0.9, 0.75, 0.6, 0.5, 0.4, 0.25, 0.1, 0.01\}$. For each α^0, record the final minimizer, final objective function value and number of iterations to terminate, for the gradient descent algorithm (without scaling) with backtracking line search and the gradient descent algorithm (with scaling) with backtracking line search. Prepare a plot where the number of iterations for both the algorithms are plotted against α^0 values. Use different colors and a legend to distinguish the plots corresponding to the different algorithms. Comment on the observations. Comment about the minimizers and objective function values obtained for different choices of the α^0 values for the two algorithms.

10. [**R**] With starting point $\mathbf{x}^0 = (1, 2000)$ and $\tau = 10^{-9}$, we will now study the behavior of gradient descent algorithm (without scaling) with backtracking line search, gradient descent algorithm (with scaling) with backtracking line search, for different choices of ρ. Take $\alpha = 1, \gamma = 0.5$. Try $\rho \in \{0.9, 0.8, 0.75, 0.6, 0.5, 0.4, 0.25, 0.1, 0.01\}$. For each ρ, record the final minimizer, final objective function value and number of iterations to terminate, for the gradient descent algorithm (without scaling) with backtracking line search and the gradient descent algorithm (with scaling) with backtracking line search. Prepare a plot where the number of iterations for both the algorithms are plotted against ρ values. Use different colors and a legend to distinguish the plots corresponding to the different algorithms. Comment on the observations. Comment about the minimizers and objective function values obtained for different choices of the ρ values for both the algorithms.

Exercise 2:

Consider the function

$$q(\mathbf{x}) = 1000(x_2 - x_1^2)^2 + (2 - x_1)^2.$$

1. [**R**] Can you design a suitable diagonal matrix $\mathbf{D}^k$ for gradient descent algorithm with scaling to solve $\min_{\mathbf{x}} q(\mathbf{x})$. If you can come up with a suitable choice of $\mathbf{D}^k$, use it in the implementation of Algorithm 1 (with backtracking line search) to find the minimizer of $q(\mathbf{x})$ with starting point $\mathbf{x}^0 = (5, 5)$ and $\tau = 10^{-9}$. Comment on your observations when compared to the gradient descent (without scaling) with backtracking line search. If you cannot find a suitable choice of $\mathbf{D}^k$, explain clearly the reasons.

2. If we allow the scaling matrix $\mathbf{D}^k$ to be non-diagonal, then a natural choice turns out to be $\mathbf{D}^k = (\nabla^2 f(\mathbf{x}^k))^{-1}$. This choice of $\mathbf{D}^k$ yields the popular **Newton's method**. Implement Algorithm 1 with this choice of $\mathbf{D}^k = (\nabla^2 f(\mathbf{x}^k))^{-1}$.

3. [**R**] Based on our discussion on condition number and the derivation of the gradient descent scheme with scaling, can you identify and write down the matrix $\mathbf{Q}$ whose condition number needs to be analyzed in the new gradient scheme with scaling with $\mathbf{D}^k = (\nabla^2 f(\mathbf{x}^k))^{-1}$?

4. [**R**] With starting point $\mathbf{x}^0 = (5, 5)$ and a stopping tolerance $\tau = 10^{-9}$, find the number of iterations taken by the gradient descent algorithm (without scaling) with backtracking line search, gradient descent algorithm (with scaling) with backtracking line search. For backtracking line search, use $\alpha^0 = 1, \rho = 0.5, \gamma = 0.5$. Note the minimizer and minimum objective function value in each case. Comment on your observations. Also note the condition number of the Hessian matrix involved in the gradient descent algorithm (without scaling) with backtracking line search and the matrix $\mathbf{Q}$ involved in the gradient descent algorithm (with scaling) with backtracking line search in each iteration. Prepare a plot depicting the behavior of condition numbers in both algorithms against iterations. Use different colors and legend to distinguish the methods. Comment on your observations.