NUMERICAL OPTIMIZATION: ASSIGNMENT 2

DEADLINE: the lab on 2024.03.11

This is the second assignment to get you up to speed with Python, NumPy and Matplotlib. You can get up to 5 points. In some tasks, the colors red and green are used. If this is inconvenient for you, feel free to change them for some other two distinct colors. In the last three tasks, avoid using loops as much as possible — organizing the computations using vectors and matrices will be *a lot* faster.

1. 1 point

   (a) Do a similar plot to the one from Assignment 1, Task 4, but plot only the *setosa* and virginica types.

   (b) Add a line $y = 2x - 8$ to the plot.

   (c) Mark the *setosa* irises green if they are below the line and green if they are above it. Mark the *virginica* irises green if they are above the line and green if they are below it.

   (d) Save the plot to the file `assignment2-1.png`.

   (e) Modify the line so that the number of red points is reduced. Save the plot to the file `assignment2-2.png`.

2. 1 point Write a program which generates a set of two-dimensional point clouds such that:

   (a) each point cloud has 1,000 (x, y) points, where x, y are given by normal distributions $N(a_i, 1)$ and $N(b_i, 1)$, respectively;

   (b) the point cloud centers (a_i, b_i) form a regular polygon with the side lendth d.

   Run the program for all cases $K \in \{7, 11, 23\}$ and $d \in \{5, 10, 15\}$. Save all the plots.

3. 1 point For the data from the previous task (choose any K), compute the distance of each data point from each point cloud center. Color red the data points which are closer to the other point cloud center than their original point cloud. Color the remaining points green. What is the relation between d and the percentage of red points? What value of d would you expect to give ca. 10% red points?

4. 1 point The matrices X, Y of sizes $d \times N$ and $d \times M$, respectively, contain N and M d-dimensional column vectors. Write a function which computes the distance matrix D of size $N \times M$ such that d_{ij} is the distance between the ith vector of matrix X and the jth vector of matrix Y. Measure the elapsed time for randomly generated matrices for $d = 100$, $N = M = 1,000$ and for $d = 100$, $N = 10,000$, $M = 1,000$.

5. 1 point The matrices X, Y are defined as in the previous task. Write a function which, for every vector in X, finds its nearest neighbor in the matrix Y. More precisely, the function has to return a $1 \times N$ vector which holds the IDs of the nearest neighbors. Write another variant of the function which returns a $k \times N$ matrix containing the IDs of k nearest neighbors (ordered by distance).