Homework 1

ELEC-E5431 – Large-Scale Data Analysis (LSD Analysis)

Return your report preferable in pdf-format by **Jan. 24, 2020** in person in class or through MyCourse or to sergiy.vorobyov@aalto.fi. Enclose you MATLAB codes as well!!! In subject line: write ELEC-E5431.

Show all the steps of your work. It is preferable if you code in MATLAB.

The optimization problem to be addressed is a simple quadratic function minimization, that is,

$$\min_{\mathbf{x}} \ \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x}$$

where the matrix $\mathbf{A}$ and vector $\mathbf{b}$ are appropriately generated. Use the same $\mathbf{A}$ and $\mathbf{b}$ throughout.

Hints:

- $\mathbf{A}$ should be such that $\mathbf{x}^T\mathbf{A}\mathbf{x}$ is non-negative. $\mathbf{b}$ should be in the range of $\mathbf{A}$.
- In fact, by solving the above unconstrained minimization problem, you solve a system of linear equations $\mathbf{A}\mathbf{x} = \mathbf{b}$. Indeed, the gradient of the objective function is $\mathbf{A}\mathbf{x} - \mathbf{b}$, and it should be equal to 0 at optimality. Thus, you can find optimal $\mathbf{x}^*$ using back-slash (or matrix inversion followed by computing the product $\mathbf{A}^{-1}\mathbf{b}$) operators in MATLAB. The optimal objective value can be then obtained by simly substituting such $\mathbf{x}$ into the objective of the above optimization problem.

Let the dimension of $\mathbf{x}$ be 100 variables or few 100's, but also play with higher dimensions. Up to which dimension MATLAB can still inverte a matrix?

Set the tolerance parameter for the stopping criterion for checking the convergence to 10^{-5}, i.e., check if $\|\nabla f(\mathbf{x})\| \leq 10^{-5}$, and limit the total number of iterations by 1000 if the predefined tolerance is still not achieved.

Problem 1: *Gradient Descent Algorithm.*

Implement Gradient Descent Algorithm and use it to solve the above optimization problem. Use correctly selected fixed step size α and the parameter γ. Draw the experimental convergence rate, i.e., draw the plot $\log \|\mathbf{x}_k - \mathbf{x}^*\|_2$ versus iteration number k, for the algorithm and compare it with the theoretically predicted one.

Problem 2: *Conjugate Gradient Algorithm.* Implement Conjugate Gradient Algorithm and use it to solve the above optimization problem. Use correctly selected fixed step size. Draw the experimental convergence rate, i.e., draw the plot $\log \|\mathbf{x}_k - \mathbf{x}^*\|_2$ versus iteration number k, for the algorithm and compare it with the theoretically predicted one.

Problem 3: *Nesterov's Algorithm.* Implement Nesterov's Algorithm and use it to solve the above optimization problem. Use correctly selected fixed step size. Draw the experimental convergence rate, i.e., draw the plot $\log \|\mathbf{x}_k - \mathbf{x}^*\|_2$ versus iteration number k, for the algorithm and compare it with the theoretically predicted one.

Problem 4: *Stochastic Coordinate Descent.* Implement Stochastic versions of the Coordinate Descent method. What is the relevance of Stochastic Coordinate Descent with Stochastic Gradient? Use it to solve the above optimization problem. Use correctly selected fixed step size. Draw the experimental convergence rate, i.e., draw the plot $\log \|\mathbf{x}_k - \mathbf{x}^*\|_2$ versus iteration number k, for the algorithm.

Problem 5: *Comparisons.* Compare the results (in terms of the iterations required and the overall computation time) for different methods (including the standard MATLAB back-slash operator) and draw your overall conclusions.