ELEC531/ELEC431 — Handout #3

Adaptive Signal Processing — Tuesday, November 2, 2021

Homework Set #3

Due: Friday, November 12, 2021.

Instructions:

For each question prepare separate Matlab/Python/Julia files (similar to previous homeworks):

In your reports, for each part of the question generate a different section. Make sure that you put introductory, descriptive and explanatory statements as well as equations formatted in Latex (if needed) for each section, such that it is organized as a full report. Do not zip your files. Submit all your files separately. If you are submitting a Python or Julia notebook, please also submit its pdf version too.

1. *Quadratic Minimization*

Consider the quadratic minimization problem with the least squares cost function:

$$f(\mathbf{x}) = \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^2$, $\mathbf{A}\mathbf{x} \in \mathbb{R}^{N\times 2}$ and $\mathbf{b} \in \mathbb{R}^N$. Matrices $\mathbf{A}$ and $\mathbf{b}$ are known and provided in our class blackboard site.

In this problem, we explore the performance of the gradient descent algorithms. Since this is the famous least squares problem, we have closed form expression for the solution:

$$\mathbf{x}_* = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}. \tag{2}$$

For this problem

(a) Find and display the expressions of the gradient $\nabla f(\mathbf{x})$ and the Hessian $\nabla^2 f(\mathbf{x})$.

(b) Is this a strongly convex function? What is the Liphschitz constant? Show the details of your answer including your Matlab computations.

(c) Write down the expression for the gradient update rule with step size μ. Describe how you would choose the step size by providing explanations and Matlab computation details.

(d) Perform gradient descent iterations with the step size you selected in the previous part. You can set your number of iterations to 100. Record your search vector values as well as the corresponding cost function value for each iteration.

(e) Perform Armijo's line search based gradient descent iterations. You can set parameter $\gamma = 10^{-3}$. Again record your search vector values as well as the corresponding cost function value for each iteration.

(f) Perform Nesterov's accelerated gradient approach. Record your search vector values as well as the corresponding cost function value for each iteration.

(g) Generate a convergence path figure that shows

- the contour lines of the cost functions,
- the convergence path of each algorithm (the consecutive points visited connected with lines)

in the 2-D domain of the function. Properly put legend, axis labels and caption to your figure.

(h) Generate a function value convergence graph showing $f(x^{(k)}) - f^*$ as a function of iterations, where f^* is the optimal value. Use logarithmic scale in the function (y) axis. Display results for all algorithms on the same plot. Properly add legend, axis labels and caption to your figure.

2. *Logistic Regression Optimization*

In this question, we explore gradient search algorithms for the Logistic Regression cost. Logistic Regression aims to place a hyperplane between two linearly separable sample sets. Figure 1 illustrates such sets with different colored samples.

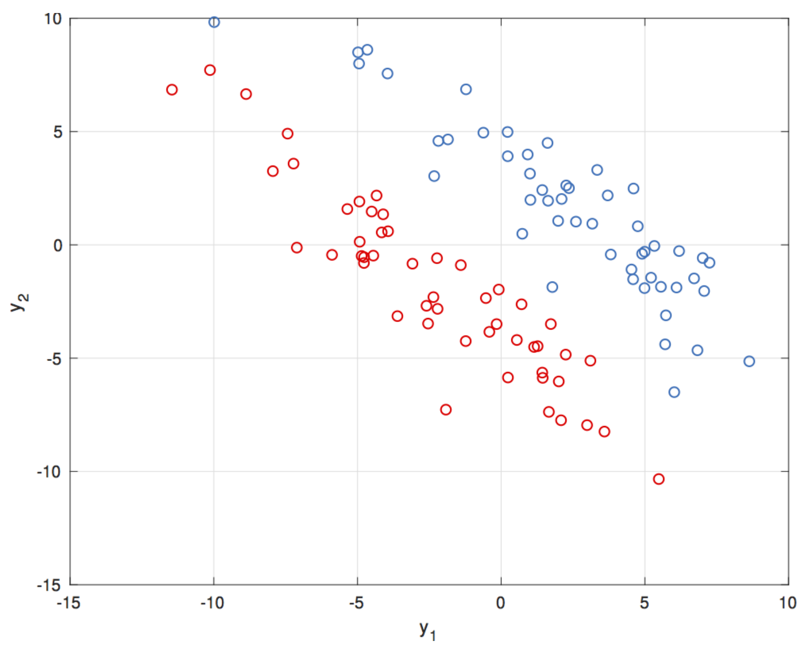

Figure 1: Example of two sets (red and blue) which are (almost) linearly separable.

The logistic regression cost function is given by

$$f(\mathbf{x}) = \sum_{k=1}^{m} \log(1 + e^{-b_k \mathbf{x}^T \mathbf{y}_k}) + \lambda \|\mathbf{x}\|_2^2 \tag{3}$$

where

- m: is the number of data points (sum of two sets in Figure 1).

- $\mathbf{y}_k$: is the vector representing the k^{th} sample point.
- $b_k \in \{-1, 1\}$: is the label of k^{th} sample point, where -1 is for one set and 1 for the other.
- $\mathbf{x}$: is the optimization variable corresponding to the normal vector of the hyperplane.

We take $\lambda = 8$ for this problem.

From (3), we can see that the cost function is the sensitive to the angle between the chosen $\mathbf{x}$ and the sample points. (Although we do not really care at the moment about the meaningfulness of the cost function $\ddot\smile$).

You can download the data samples and their labels from our course blackboard site.

Here is the task:

(a) Find and display the expressions of the gradient $\nabla f(\mathbf{x})$ and the Hessian $\nabla^2 f(\mathbf{x})$.

(b) Is this a strongly convex function? Provide your justification.

(c) Write down the expression for the gradient update rule with constant step size μ.

(d) Perform gradient descent iterations with the step size $\mu = 0.01$. You can set your number of iterations to 100. Record your search vector values as well as the corresponding cost function value for each iteration.

(e) Apply Newton algorithm with line search. You can set line search parameter as $\gamma = 10^{-4}$. Record your search vector values as well as the corresponding cost function value for each iteration.

(f) Perform Nesterov's accelerated gradient approach. Record your search vector values as well as the corresponding cost function value for each iteration.

(g) Generate a convergence path figure that shows

- the contour lines of the cost functions,
- the convergence path of each algorithm (the consecutive points visited connected with lines)

in the 2-D domain of the function. Properly put legend, axis labels and caption to your figure.

(h) Generate a function value convergence graph showing $f(x^{(k)}) - f^*$ as a function of iterations, where f^* is the optimal value. Use logarithmic scale in the function (y) axis. Display results for all algorithms on the same plot. Properly add legend, axis labels and caption to your figure. (for this problem f^* has no analytical formula, you can pick it as the last iteration value of the Newton algorithm).

(i) Now as the final graph, show sample points as in Figure 1, and place the hyperplane with the normal vector corresponding to the Newton algorithm's solution.