Nonlinear Optimization I, Fall 2021
Homework 2
Due before lecture on 9/30 (NOTE: due date delayed by two days)

Your submitted solutions to homeworks should be entirely your own work. Do not copy solutions from other students or any online source. You are allowed to discuss homework problems at a high-level with other students, but should carry out the execution of any thoughts/directions discussed independently, on your own. Feel free to cite any result presented in class without proof.

You can write solutions by hand or type them up (the LaTeX code for this pdf is on blackboard).

Q1. Consider the following differentiable objective functions for any $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$

(a) Show $f(x) = \frac{1}{2}\|Ax-b\|_2^2$ has $\lambda_{max}(A^TA)$-Lipschitz gradient and is $\lambda_{min}(A^TA)$-strongly convex.

(b) Show the potentially nonconvex function $f(x) = \frac{\exp(c^Tx)}{1+\exp(c^Tx)}$ has a Lipschitz gradient[1].

Q2. : Consider any convex $f\colon \mathbb{R}^d \to \mathbb{R}$ (although potentially not differentiable). Fix any $\bar{x} \in \mathbb{R}^d$ and suppose $\bar{x}^* \in \mathbb{R}^d$ globally minimizes

$$\min_{x\in\mathbb{R}^d} f(x) + \frac{\rho}{2}\|x-\bar{x}\|_2^2$$

for some $\rho > 0$. Prove that f is lower bounded by the following linear function for all $x \in \mathbb{R}^d$

$$f(x) \geq f(\bar{x}^*) + g^T(x-\bar{x}^*)$$

where $g = \rho(\bar{x}-\bar{x}^*)$ (that is, show that $g \in \partial f(\bar{x}^*)$ is a subgradient of f at $\bar{x}^*$).

Q3. In lecture, we saw that gradient descent can converge faster for μ-strongly convex functions. In this exercise, you will show that a modification of the accelerated method speeds up similarly.

(a) For any μ-strongly convex f (although potentially not differentiable) with global minimizer x^*, show that every $x \in \mathbb{R}^d$ has

$$f(x) \geq f(x^*) + \frac{\mu}{2}\|x-x^*\|_2^2.$$

(b) In particular, $f(x_0) - f(x^*) \geq \frac{\mu}{2}\|x_0-x^*\|_2^2$. Assuming f has L-Lipschitz gradient, conclude that after $k = \left\lceil 4\sqrt{L/\mu}\right\rceil$ iterations, the accelerated method analyzed in lecture must have

$$f(y_k) - f(x^*) \leq (f(y_0) - f(x^*))/2.$$

(c) Inspired by this geometric decrease, we can define a "Restarted Accelerated Method" that runs the accelerated method for $\lceil 4\sqrt{L/\mu}\rceil$ steps and then restarts itself, running the accelerated method with $x_0 \leftarrow y_{\lceil 4\sqrt{L/\mu}\rceil}$. Prove that for any $0 < \epsilon < f(x_0) - f(x^*)$, such a restarting algorithm will find some y with $f(y) - f(x^*) \leq \epsilon$ after a total number of iterations at most

$$\lceil 4\sqrt{L/\mu}\rceil \left\lceil \log_2\left(\frac{f(x_0)-f(x^*)}{\epsilon}\right)\right\rceil.$$

[1]The best Lipschitz constant for the gradient of $f(x) = \frac{\exp(c^Tx)}{1+\exp(c^Tx)}$ is $\|c\|_2^2/6\sqrt{3}$, but for this question it suffices for you to prove Lipschitzness with any constant you can.

Q4. Consider the following "least squares" optimization problem[2]

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|_2^2$$

for some $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. For this exercise, first write a program that generates A and b with i.i.d. normally distributed entries (mean zero, variance one) and $n = 1000$, $m = 2000$.

(a) Compute and print out the Lipschitz gradient constant and strong convexity constant for your randomly generated least squares problem instance using the results of Q1(a).

(b) Implement and run 100 steps of Gradient Descent on your randomly generated problem instance using $x_0 = 0$ and the theoretically justified stepsize $\alpha_k = 1/\lambda_{max}(A^T A)$. Print out $\|\nabla f(x_k)\|$ each iteration.

(c) Implement and run 100 steps of the Accelerated Gradient Method using $x_0 = 0$ and the same stepsize $\alpha_k = 1/\lambda_{max}(A^T A)$. Print out $\|\nabla f(x_k)\|$ each iteration. Does this outperform (b)?

(d) Implement and run 100 steps of the Restarted Accelerated Gradient Method discussed in Q3 (every 25 iterations, restart the above accelerated method, initialized at its last iterate). Print out $\|\nabla f(x_k)\|$ each iteration. Does this outperform (c)?

General Guidelines for Programming HW Problems: You can do programming assignments in any programming language you feel comfortable with (python, matlab, java, c/c++, haskell, etc). Programming questions will ask for you to solve a particular problem or describe particular settings to run an algorithm under. You must submit both your code and the requested output/plots from running your code. Grading will focus primarily on the quality of these outputs rather than of your code.

[2]You may recall our historical motivating example from lecture of fitting an ellipse to observed locations of the Ceres asteroid/dwarf planet took this form with $n = 3$ and m equal to the number of past observations.