Nonlinear Optimization I, Fall 2021
Homework 4
Due before lecture on 11/9
Please only in file formats .pdf, .zip, standard image formats (.jpg, .png, etc.).

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. The similarities between our $O(1/\sqrt{T})$ result for deterministic nonsmooth optimization and $O(1/\sqrt{T})$ result for stochastic smooth optimization extend deeply. In this question, you'll show a $O(1/\sqrt{T})$ convergence rate for stochastic nonsmooth optimization.
Consider minimizing a convex function $f(x)$, given a *stochastic subgradient oracle* $g(x)$, such that

$$\mathbb{E}[g(x)] \in \partial f(x) \quad \text{and} \quad \mathbb{E}[\|g(x)\|^2] \leq M^2,$$

via the following Stochastic Subgradient Method with $\alpha_k > 0$:

$$x_{k+1} = x_k - \alpha_k g(x_k) \ .$$

(a) Derive the following inequality bounding the expected change in distance to a minimizer x^* from one-step of this method for fixed x_k

$$\mathbb{E}\left[\|x_{k+1} - x^*\|^2 \mid x_k\right] \leq \|x_k - x^*\|^2 - 2\alpha_k(f(x_k) - f(x^*)) + \alpha_k^2 M^2.$$

(b) Use this to provide any upper bound on $\mathbb{E}\left[\min_{i\leq k}\{f(x_i) - f(x^*)\}\right]$ for any sequence α_k.

(c) For some fixed k, propose a sequence α_i such that your bound after k steps is at most $O(1/\sqrt{k})$.[1]

(d) Propose a sequence α_i such that your bound after k steps is at most $O(\log(k)/\sqrt{k})$ for all k.

Q2. Consider a continuously differentiable $F\colon \mathbb{R}^d \to \mathbb{R}^d$ where $F(x)$ is L-Lipschitz and bounded uniformly by $\|F(x)\| \leq M$ and has Jacobian $\nabla F(x)$ Q-Lipschitz and bounded uniformly by $\|\nabla F(x)\| \leq N$. Rather than searching for a solution to the nonlinear system of equations

$$F(x) = 0$$

as Newton's Method does, we could instead solve the nonlinear optimization problem

$$\min_{x\in\mathbb{R}^d} h(x) := \frac{1}{2}\|F(x)\|_2^2$$

(which of course has minimum value zero attained at the solutions above).

[1] For (c) and (d), you only need to show that your upper bound has dependence on k matching the claimed order of magnitude $O(1/\sqrt{k})$ or $O(\log(k)/\sqrt{k})$. It can have other constants like M or $\|x_0 - x^*\|$ occurring freely.

(a) Derive a formula for the gradient $\nabla h(x)$ and compute a Lipschitz constant for $\nabla h(x)$.

(b) Consider running gradient descent here $x_{k+1} = x_k - \alpha_k \nabla h(x_k)$. How does the per iteration cost of this compare to that of Newton's method? Give an example where this method converges to a point x^* with $F(x^*) \neq 0$ despite points with $F(x) = 0$ existing.

(c) **(Bonus 1pt)** Suppose $F = \nabla f$ for some function we are interested in minimizing. Devise a generic condition relating the gradient and Hessian of f under which you can prove the convergence of this method in terms of $\|\nabla f(x_k)\| \to 0$.

Q3. Let $A \in \mathbb{R}^{n\times n}$ be a real symmetric matrix.

(a) Write down a formula for Newton's Method applied to the system $n+1$ equations

$$(A - \lambda I)x = 0 \quad \text{and} \quad x^T x = 1$$

with $n+1$ unknowns (x, λ). Using this formula, write a program that applies Newton's Method to find and print out an eigenpair (x, λ) for the matrix

$$A = \begin{bmatrix} 4 & 2 & 1 \\ 2 & 3 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad \text{from initial point } x_0 = \begin{bmatrix} 1/5 \\ -1/5 \\ 4/5 \end{bmatrix} \text{ and } \lambda_0 = 1.$$

(b) Using the gradient descent algorithm proposed in Q2(b), write a program that finds and prints out an eigenpair (x, λ) for the above example matrix and initialization. Use a backtracking Armijo linesearch initialized with $\alpha = 100$, $\tau = 0.9$, $\eta = 0.1$. How does this method's performance compare to Newton's Method?

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.