CS 726: Homework #3

Posted: Feb 18, 2025. Due: Mar 10, 2025 on Canvas

Please typeset your solutions.

You should provide sufficient justification for the steps of your solution. The level of detail should be such that your fellow students can understand your solution without asking you for further explanation.

Q1	Q2	Q3.1	Q3.2	Q4	Q5	Q6	Total
10	10	10	10	15	10	35	100 pts

Note: You can use the results we have proved in class – no need to prove them again (unless you are explicitly asked to prove them).

Q 1

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a continuously differentiable function. In Homework #2, we proved that if f is convex and L-smooth with respect to the Euclidean norm $\|\cdot\|_2$, then

$$(\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d) : \langle \nabla f(\mathbf{x}) - \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \geq \frac{1}{L} \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2^2 \tag{1}$$

Prove the converse: If the inequality in Eq. (1) holds, then f is convex and L-smooth.

Hint: Recall the first-order condition for convexity: $f(\mathbf{w}) \geq f(\mathbf{z}) + \nabla f(\mathbf{z})^T (\mathbf{w} - \mathbf{z}), \forall \mathbf{w}, \mathbf{z}$.

Solution:

YOUR SOLUTION HERE

Q 2 Monotonicity of GD in gradient norm

Use (1) to prove that for an L-smooth convex function f, standard gradient descent with step size $\alpha = \frac{1}{L}$ guarantees that $\|\nabla f(\mathbf{x}_{k+1})\|_2 \leq \|\nabla f(\mathbf{x}_k)\|_2$, for all $k \geq 0$.

Solution:

YOUR SOLUTION HERE

In Q3 and Q4, we will show that up to log factors, one can reduce the problem of minimizing a non-strongly convex function to that of minimizing a strongly convex function.

Q 3 Strongly convex regularizer

Consider the unconstrained optimization problem $\min_{x \in \mathbb{R}^d} f(x)$, where f is an L-smooth and convex function with a minimizer $x^* \in \arg\min_{x \in \mathbb{R}^d} f(x)$.

Let $x_0 \in \mathbb{R}^d$ be a given point that satisfies $\|x_0 - x^*\|_2 \leq R$ for some $R \in (0, \infty)$. Given a target error $\epsilon > 0$, define a new function $f_\epsilon(x) = f(x) + \frac{\epsilon}{2R^2} \|x - x_0\|_2^2$, which is a regularized version of f.

Q 3.1

Prove that f_ϵ, as a function of x, is $\left(L + \frac{\epsilon}{R^2}\right)$-smooth and $\frac{\epsilon}{R^2}$-strongly convex.

Solution:
YOUR SOLUTION HERE

Q 3.2

Prove that

$$\forall x \in \mathbb{R}^d : \quad f(x) - f(x^*) \leq f_\epsilon(x) - f_\epsilon(x_\epsilon^*) + \frac{\epsilon}{2}, \tag{2}$$

where $x_\epsilon^* := \arg\min_{x \in \mathbb{R}^d} f_\epsilon(x)$ is the minimizer of the regularized problem. (This means if x is near-optimal for the regularized problem f_ϵ, then it is also near-optimal for the original f.)

Solution:
YOUR SOLUTION HERE

Q 4 Strongly convex regularizer, continued

Consider the same setting as Q3 and assume in addition that ϵ satisfies $\epsilon \leq LR^2$. Below GD=standard gradient descent, and AGD=Nesterov's accelerated gradient descent.

Q 4.1

Use the convergence bound for GD for smooth and *strongly* convex functions we derived in class to prove the following: If we apply GD to the strongly convex function f_ϵ with initial solution x_0, GD finds a point x_k satisfying $f(x_k) - f(x^*) \leq \epsilon$ in at most $O\left(\frac{LR^2}{\epsilon} \log \frac{LR^2}{\epsilon}\right)$ iterations.

How does this compare to applying GD directly to f and using the convergence bound for GD and smooth and (non-strongly) convex functions?

Solution:
YOUR SOLUTION HERE

Q 4.2

Use the convergence bound for AGD for smooth and *strongly* convex functions we derived in class to prove the following: If we apply AGD to the function f_ϵ with initial solution x_0, AGD finds a point x_k satisfying $f(x_k) - f(x^*) \leq \epsilon$ in at most $O\left(\sqrt{\frac{LR^2}{\epsilon}} \log \frac{LR^2}{\epsilon}\right)$ iterations.

How does this compare to applying AGD directly to f and using the AGD bound for smooth and (non-strongly) convex functions?

Solution:
YOUR SOLUTION HERE

Q 5 Sharpness and geometric convergence

Recall that gradient descent when applied to an L-smooth convex function f with step size $\alpha_k = \frac{1}{L}$ guarantees that at iteration $k \geq 1$,

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{L\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{2k}, \tag{3}$$

where $\mathbf{x}^*$ is a minimizer of f.

Suppose in addition that our function f is $(2, \mu)$-sharp. That is,

$$(\forall \mathbf{x} \in \mathbb{R}^d): \quad f(\mathbf{x}) - f(\mathbf{x}^*) \geq \frac{\mu}{2} \mathrm{dist}(\mathbf{x}, \mathcal{X}^*)^2, \tag{4}$$

where $\mathcal{X}^*$ is the set of all minimizers of f and $\mathrm{dist}(\mathbf{x}, \mathcal{X}^*) = \inf_{\mathbf{y} \in \mathcal{X}^*} \|\mathbf{x} - \mathbf{y}\|$.

Prove using (3) and (4) that after at most $k = \lceil \frac{2L}{\mu} \rceil$ iterations, we have

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{f(\mathbf{x}_0) - f(\mathbf{x}^*)}{2}. \tag{5}$$

Argue using (5) that for any $\epsilon > 0$, gradient descent guarantees

$$f(\mathbf{x}_K) - f(\mathbf{x}^*) \leq \epsilon$$

after at most $K = O\big(\frac{L}{\mu} \log\big(\frac{f(\mathbf{x}_0) - f(\mathbf{x}^*)}{\epsilon}\big)\big)$ iterations.

(This question shows that geometric convergence can be achieved under sharpness, a less restrictive condition than strong convexity.)

Solution:

YOUR SOLUTION HERE

Q 6 Coding Assignment

You should code in Python 3.7+ and your code needs to compile/run without any errors to receive any points for the coding assignment. Jupyter notebook is accepted and you may use the provided **hw3_template.ipynb** as a starting point for the first coding assignment. You may also start from scratch, but the plots should meet the requirements. You may only use modules from the Python standard library plus NumPy and Matplotlib.

Your submission for Homework #3 should be two files: **hw3.ipynb** (or **hw3.py**) and **hw3.pdf**, and do **NOT** archive into a zip file.

- The .ipynb or .py file should implement the algorithms and produce the required figures.
- In the .pdf, you should include the answers to the questions below **AND the figures produced by your python code.**

In this question, you are asked to implement the following four algorithms:

1. Gradient descent (GD) with a constant step size $\alpha_k \equiv 1/L$.
2. Gradient descent with the exact line search. In every step, this method sets the step size α_k as

$$\alpha_k = \operatorname*{argmin}_{\alpha \in \mathbb{R}} f(\mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)).$$

3. Lagged steepest descent, defined as follows: Let α_k be the exact line search gradient descent step size corresponding to the point $\mathbf{x}_k$. Lagged steepest descent updates the iterates as: $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_{k-1} \nabla f(\mathbf{x}_k)$ (i.e., the step size "lags" by one iteration), with the initial value

$$\alpha_{-1} = 1/L.$$

4. Nesterov's accelerated gradient descent (AGD) for smooth convex minimization.

The problem instance we will consider here is $\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})$, where $d = 250$ and $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{M}\mathbf{x} - \mathbf{b}^T\mathbf{x}$ is a quadratic function given by:

$$\mathbf{M} = \begin{bmatrix} 2 & -1 & 0 & 0 & \dots & 0 & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 2 & -1 & \dots & 0 & 0 \\ 0 & 0 & -1 & 2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & -1 & 2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

It can be shown that the top eigenvalue of $\mathbf{M}$ is $L = 4$, which you should use for setting the step sizes of GD and AGD. Also note that for this quadratic function f, the argmin in exact line search can be computed in closed form.

Initialize all algorithms at $\mathbf{x}_0 = \mathbf{0}$ and run 2500 iterations. All your plots should be showing the optimality gap $f(\mathbf{x}_k) - f(\mathbf{x}^*)$ on the y-axis and the iteration count on the x-axis. The y-axis should be shown on a logarithmic scale. Note that by setting the gradient to zero, you can directly compute the minimizer $\mathbf{x}^*$ of f given $\mathbf{M}$ and $\mathbf{b}$, and thus you can also compute $f(\mathbf{x}^*)$.

Q 6.1

Use a single plot to compare gradient descent with constant step size, gradient descent with the exact line search, and Nesterov's algorithm. Use different colors for different algorithms and show a legend with descriptive labels (e.g., GD:constant, GD:exact, and Nesterov). Discuss the results. Which method is faster? Are the function values monotonically decreasing? Are the behaviors of the algorithms consistent with the theoretical results you saw in class?

Q 6.2

Use a single plot to compare Nesterov's algorithm to lagged steepest descent. You should, again, use different colors and a legend. Compare the two algorithms and discuss the results in a similar way as in the previous part.

Q 6.3

Modify the output of Nesterov's algorithm and lagged steepest descent: you should still run the same algorithms, but for each of the two algorithms, your plot at each iteration k should show $\min_{0\le i\le k} f(\mathbf{x}_i) - f(\mathbf{x}^*)$, that is, the lowest function value up to iteration k minus the optimal value. Discuss the results.

Solution:

YOUR SOLUTION HERE