CS 726: Homework #4

Posted: Mar 31, 2025. Due: Apr 13, 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.

Note: You can use the results we have proved in class – no need to prove them again.

Q1.1	Q1.2	Q1.3	Q2.1	Q2.2	Q2.3	Q3.1	Q3.2	Q4.1	Q4.2	Total
10	10	10	5	10	10	20	10	7	8	100 pts

Q 1

Let $f : \mathbb{R}^d \to \mathbb{R}$ be an L-smooth function and let $\mathcal{X} \subseteq \mathbb{R}^d$ be a closed, convex, and nonempty set. Recall the definition of the gradient mapping: $G_\eta(\mathbf{x}) = \eta\left(\mathbf{x} - P_{\mathcal{X}}\left(\mathbf{x} - \frac{1}{\eta}\nabla f(\mathbf{x})\right)\right)$, where $P_{\mathcal{X}}(\cdot)$ denotes the Euclidean projection onto $\mathcal{X}$.

Q 1.1

Prove that $G_\eta(\cdot)$ is $(2\eta + L)$-Lipschitz continuous, that is, $\|G_\eta(\mathbf{x}) - G_\eta(\mathbf{y})\|_2 \leq (2\eta + L)\|\mathbf{x} - \mathbf{y}\|_2, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.

Solution:

Q 1.2

Now assume in addition that f is convex and define: $T(\mathbf{x}) = \mathbf{x} - \eta\nabla f(\mathbf{x})$, where $0 \leq \eta \leq \frac{1}{L}$. Prove that $\|T(\mathbf{x}) - T(\mathbf{y})\|_2 \leq \|\mathbf{x} - \mathbf{y}\|_2, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.

Hint: It might be helpful to use the inequality in HW#3 Q1.

Solution:

Q 1.3

Again assume that f is convex. Let $\mathbf{x}_k$ be the kth iterate of projected gradient descent with step size $1/L$ applied to f. Use Part 2 of this question to prove the following inequality:

$$(\forall k \geq 0): \quad \|G_L(\mathbf{x}_{k+1})\|_2 \leq \|G_L(\mathbf{x}_k)\|_2. \tag{1}$$

Solution:

Q 2

Consider a constrained minimization problem $\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$, where $f : \mathbb{R}^d \to \mathbb{R}$ is L-smooth and $\mathcal{X} \subseteq \mathbb{R}^d$ is a hyper-rectangle: $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : x_i \in [a_i, b_i], \forall i \in \{1, 2, \ldots, d\}\}$, where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$ are fixed vectors that satisfy $\forall i \in \{1, \ldots, d\} : a_i < b_i$.

Q 2.1

We claimed in class that $P_{\mathcal{X}}(\mathbf{x}) = \max\{\mathbf{a}, \min\{\mathbf{x}, \mathbf{b}\}\}$, where min and max operations are applied element-wise. Prove that this is indeed a correct expression for the projection operator in this case.

Solution:

Q 2.2

For each $i \in \{1, \dots, d\}$, define the shorthand $\Delta_i(\mathbf{x}) := \nabla_i f(\mathbf{x})\mathbf{e}_i$, where $\mathbf{e}_i \in \mathbb{R}^d$ is the i^{th} standard basis vector (the i^{th} coordinate of $\mathbf{e}_i$ equals one; all other coordinates are equal to zero). Define:

$$T(\mathbf{x}, i) := \operatorname*{argmin}_{\mathbf{u}\in\mathcal{X}} \left\{ \langle \Delta_i(\mathbf{x}), \mathbf{u} - \mathbf{x} \rangle + \frac{L}{2}\|\mathbf{u} - \mathbf{x}\|_2^2 \right\}. \tag{2}$$

Express the gradient mapping $G_L(\mathbf{x})$ in terms of L, $\mathbf{x}$, and $T(\mathbf{x}, i)$, $i \in \{1, \dots, d\}$.

Solution:

Q 2.3

Consider the following iterative method that starts from some initial point $\mathbf{x}_0 \in \mathcal{X}$ and updates its iterates $\mathbf{x}_k$ for $k \geq 0$ as:

$$\begin{aligned} i_k^* &= \operatorname*{argmax}_{1\leq i\leq d} |(G_L(\mathbf{x}_k))_i|, \\ \mathbf{x}_{k+1} &= T(\mathbf{x}_k, i_k^*). \end{aligned}$$

Prove the following sufficient descent property for this algorithm:

$$(\exists \alpha > 0)(\forall k \geq 0): \quad f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) - \frac{\alpha}{2}\|G_L(\mathbf{x}_k)\|_2^2. \tag{3}$$

What is the largest α for which this property holds? What can you say about convergence of this method if f is bounded below by some number f_*?

Solution:

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.

Your submission for Homework #4 should be two files: **hw4.ipynb** (or **hw4.py**) and **hw4.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.**

For the coding assignment, you should implement the following four methods:

1. The method of conjugate gradients with Dai-Yuan rule (which is equivalent to the one we derived in class):

$$\mathbf{x}_{k+1} = \mathbf{x}_k - h_k\mathbf{p}_k, \quad \text{where } h_k = \arg\min_{h\in\mathbb{R}} f(\mathbf{x}_k - h\mathbf{p}_k). \tag{4}$$

$$\beta_k = \frac{\|\nabla f(\mathbf{x}_{k+1})\|_2^2}{\langle \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k), \mathbf{p}_k \rangle}, \tag{5}$$

$$\mathbf{p}_{k+1} = \nabla f(\mathbf{x}_{k+1}) - \beta_k \mathbf{p}_k, \tag{6}$$

with initial search direction $\mathbf{p}_0 = \nabla f(\mathbf{x}_0)$.

2. The Heavy-Ball method, which applies to L-smooth and m-strongly convex functions, and whose updates are defined by:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_1 \nabla f(\mathbf{x}_k) + \alpha_2(\mathbf{x}_k - \mathbf{x}_{k-1}),$$

where $\alpha_1 = \frac{4}{(\sqrt{L}+\sqrt{m})^2}$ and $\alpha_2 = \frac{\sqrt{L}-\sqrt{m}}{\sqrt{L}+\sqrt{m}}$.

3. Nesterov's AGD method for L-smooth and m-strongly convex optimization (Lecture 9–10, Algorithm 1).

4. Nesterov's AGD method for L-smooth convex optimization (Lecture 9–10, Algorithm 2; you already implemented this in HW3).

Q 3

The problem instance we will consider is $\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})$, where $d = 150$, $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{M}\mathbf{x} - \mathbf{b}^T\mathbf{x} + \frac{m}{2}\|\mathbf{x}\|_2^2$ and

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

Note that $\mathbf{M}$ and $\mathbf{b}$ can be generated using the same code as in the last homework. The function f is m-strongly convex and $L = 4 + m$ smooth.

Q 3.1

Write the code that implements the four methods above: 1. conjugate gradient method, 2. the heavy ball method. 3. Nesterov's AGD for smooth and strongly convex minimization, and 4. Nesterov's AGD for smooth minimization. Your code should produce three figures, corresponding to three different values of the strong convexity parameter m: (1) $m = 1$, (2) $m = 0.1$, and (3) $m = 0.01$. Each figure should contain four curves, showing the optimality gap $f(\mathbf{x}_k) - f(\mathbf{x}^*)$ (on a logarithmic scale) against the iteration count k for each of the four methods. Each run should be for 1500 iterations. The initial point for all the methods should be $\mathbf{x}_0 = \mathbf{0}$.

Discuss your results. Which one is the fastest? How do the two variants of Nesterov's AGD compare with each other? How does the result depend on m? Does CG find the optimal solution after a finite number of iterations? Are the numerical results consistent with the theory from class?

Q 3.2

Now, modify Nesterov's AGD for smooth minimization (method 4 above) by replacing $\mathbf{x}_{k+1} = \mathbf{y}_k - \frac{1}{L}\nabla f(\mathbf{y}_k)$ with

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{y}_k - \frac{1}{L}\nabla f(\mathbf{y}_k), & \text{if } f\big(\mathbf{y}_k - \frac{1}{L}\nabla f(\mathbf{y}_k)\big) \le f\big(\mathbf{x}_k - \frac{1}{L}\nabla f(\mathbf{x}_k)\big), \\ \mathbf{x}_k - \frac{1}{L}\nabla f(\mathbf{x}_k), & \text{otherwise.} \end{cases} \tag{7}$$

Produce the same set of plots as in Part 1 of the question, i.e., for the 4 methods and the three values of m.

In the discussion, you should

- Discuss the above numerical results. What has changed compared to Part 1 of the question and explain why?
- Argue that the function value over the iterates of the modified method is monotonically non-increasing.
- Write down a theoretical bound on the number of iterations needed by the modified method produce a point $\mathbf{x}_k$ with $f(\mathbf{x}_k) - f(\mathbf{x}^*) \le \epsilon$. (Hint: your bound may look like the one in the beginning of Section 2, Lecture 12. You may use the Big-Oh notation.)

Solution:

Q 4

In this part, you will compare the heavy ball method and Nesterov's method for smooth and *strongly* convex optimization. Both methods were defined in previous question, and you can, of course, reuse the code you wrote.

Your problem instance is the following one-dimensional minimization problem: $\min_{x \in \mathbb{R}} f(x)$, where

$$f(x) = \begin{cases} \frac{25}{2}x^2, & \text{if } x < 1 \\ \frac{1}{2}x^2 + 24x - 12, & \text{if } 1 \leq x < 2 \\ \frac{25}{2}x^2 - 24x + 36, & \text{if } x \geq 2. \end{cases}$$

Q 4.1

Prove that f is μ-strongly convex and L-smooth with $\mu = 1$ and $L = 25$. What is the global minimizer of f? (Justify your answer.)

Q 4.2

Run Nesterov's method and the heavy-ball method, starting from $x_0 = 3.3$. Plot the optimality gap of Nesterov's method and the heavy ball method over 150 iterations. What do you observe? What does this plot tell you?

Solution: