CS 726: Homework #3

Posted: 10/07/2020, due: 10/21/2020 by 5pm on Canvas

Please typeset or write your solutions neatly! If we cannot read it, we cannot grade it.

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

Q 1. Let $f : \mathbb{R}^d \to \bar{\mathbb{R}}$ be a differentiable function.

(i) Prove that if f is L-smooth, then:

$$(\forall \mathbf{x} \in \mathbb{R}^d): \quad \frac{1}{2L}\|\nabla f(\mathbf{x})\|_2^2 \leq f(\mathbf{x}) - f(\mathbf{x}^*) \leq \frac{L}{2}\|\mathbf{x} - \mathbf{x}^*\|_2^2. \qquad \text{[5pts]}$$

(ii) Prove that if f is convex and L-smooth, then the following holds:

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

Hint: Consider the function $f_{\mathbf{x}}(\mathbf{z}) = f(\mathbf{z}) - \langle \nabla f(\mathbf{x}), \mathbf{z} \rangle$. What is its minimizer? Is $f_{\mathbf{x}}(\mathbf{z})$ smooth? [10pts]

(iii) Prove the converse to Part (ii): If the inequality in Eq. (1) holds, then f is convex and L-smooth. [10pts]

Q 2. Consider the unconstrained optimization problem $\min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x})$, where f is an L-smooth convex function. Assume that $\|\mathbf{x}_0 - \mathbf{x}^*\|_2 \leq R$, for some $R \in (0, \infty)$, and let $f_\epsilon(\mathbf{x}) = f(\mathbf{x}) + \frac{\epsilon}{2R^2}\|\mathbf{x} - \mathbf{x}_0\|_2^2$. Let $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x})$ and $\mathbf{x}_\epsilon^* = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^d} f_\epsilon(\mathbf{x})$. Prove that:

$$(\forall \mathbf{x} \in \mathbb{R}^d): f(\mathbf{x}) - f(\mathbf{x}^*) \leq f_\epsilon(\mathbf{x}) - f_\epsilon(\mathbf{x}_\epsilon^*) + \frac{\epsilon}{2}.$$

(i) Prove that standard gradient descent applied to function f_ϵ finds a point $\mathbf{x}_k$ with $f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \epsilon$ in $O(\frac{L}{\epsilon} R \log(\frac{LR^2}{\epsilon}))$ iterations. How does this compare to applying gradient descent directly to f? [10pts]

(ii) Prove that Nesterov's method for smooth and strongly convex minimization applied to f_ϵ will find a solution $\mathbf{x}_k$ with $f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \epsilon$ in $O(\sqrt{\frac{L}{\epsilon}} R \log(\frac{LR^2}{\epsilon}))$ iterations. [5pts]

Q 3. In class, we have analyzed the following variant of Nesterov's method for L-smooth convex optimization:

$$\begin{aligned} \mathbf{x}_k &= \frac{A_{k-1}}{A_k}\mathbf{y}_{k-1} + \frac{a_k}{A_k}\mathbf{v}_{k-1} \\ \mathbf{v}_k &= \mathbf{v}_{k-1} - a_k \nabla f(\mathbf{x}_k)/L \\ \mathbf{y}_k &= \mathbf{x}_k - \frac{1}{L}\nabla f(\mathbf{x}_k), \end{aligned}$$

where L is the smoothness constant of f, $a_0 = A_0 = 1$, $\frac{{a_k}^2}{A_k} = 1$, $A_k = \sum_{i=0}^k a_i$. We take $\mathbf{x}_0 \in \mathbb{R}^d$ to be an arbitrary initial point and $\mathbf{y}_0 = \mathbf{v}_0 = \mathbf{x}_0 - \nabla f(\mathbf{x}_0)/L$.

Prove that we can state Nesterov's method in the following equivalent form:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{y}_{k-1} + \frac{a_k}{A_k}\Big(\frac{A_{k-1}}{a_{k-1}} - 1\Big)(\mathbf{y}_{k-1} - \mathbf{y}_{k-2}), \\ \mathbf{y}_k &= \mathbf{x}_k - \frac{1}{L}\nabla f(\mathbf{x}_k). \end{aligned} \qquad (2)$$

Hint: It is helpful to first prove that $\mathbf{y}_k = \frac{A_{k-1}}{A_k}\mathbf{y}_{k-1} + \frac{a_k}{A_k}\mathbf{v}_k$. [10pts]

Coding Assignment

You should code in MATLAB or Python 3.7+ and your code needs to compile/run without any errors to receive any points for the coding assignment. For Python, Jupyter notebook is accepted and you can include the discussion and figures in the Jupyter notebook. You may only use modules from the Python standard library plus NumPy and Matplotlib. Do **not** archive your code file(s) into a zip file.

You should turn in both the code (as text file(s)) and a pdf with the figures produced by your code together with the appropriate answers to the questions below.

Q 4. In this question, you are asked to implement the following algorithms:

- Steepest descent with the constant step size $\alpha_k = 1/L$.
- Steepest descent with the exact line search. In every step, this method sets the step size α_k as

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

- Lagged steepest descent, defined as follows: Let α_k be the exact line search steepest 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).
- Nesterov's method for smooth convex minimization.

The problem instance we will consider here is $\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})$, where $d = 200$ and $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{M}\mathbf{x} - \mathbf{b}^T\mathbf{x}$ is characterized by:

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

$\mathbf{M}$ and $\mathbf{b}$ can be generated in Matlab using:

```
k = d;
M = diag(2*[ones(k, 1); zeros(d-k, 1)], 0)...
    + diag([-ones(k-1, 1); zeros(d-k, 1)], -1)...
    + diag([-ones(k-1, 1); zeros(d-k, 1)], 1);
b = zeros(d, 1);
b(1) = b(1) + 1;
```

Observe that you can compute the minimizer $\mathbf{x}^*$ of f given $\mathbf{M}$ and $\mathbf{b}$, and thus you can also compute $f(\mathbf{x}^*)$. It is possible to show that the top eigenvalue of $\mathbf{M}$ is $L = 4$.

Initialize all algorithms at $\mathbf{x}_0 = \mathbf{0}$. All your plots should be showing the optimality gap $f(\mathbf{x}) - f(\mathbf{x}^*)$ (with $\mathbf{x} = \mathbf{y}_k$ for Nesterov and $\mathbf{x} = \mathbf{x}_k$ for all other methods) on the y-axis and the iteration count on the x-axis. The y-axis should be shown on a logarithmic scale (use `set(gca, 'YScale', 'log')` after the figure command in Matlab).

(i) Use a single plot to compare steepest descent with constant step size, steepest 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., SD:constant, SD:exact, and Nesterov). Discuss the results. Do you see what you expect from the analysis we saw in class?

(ii) Use a single plot to compare Nesterov's algorithm to lagged steepest descent. You should, again, use different colors and a legend. What can you say about lagged steepest descent? How does it compare to Nesterov's algorithm?

(iii) Modify the output of Nesterov's algorithm and lagged steepest descent: you should still run the same algorithms, but now your plot at each iteration k should show the lowest function value up to iteration k for each of the two algorithms. Discuss the results. [30pts]

Q 5. In this part, you will compare the heavy ball method to Nesterov's method for smooth and strongly convex optimization. The heavy ball method, which applies to L-smooth an m-strongly convex functions, is 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 $\alpha_2 = \left(\frac{\sqrt{L}-\sqrt{m}}{\sqrt{L}+\sqrt{m}}\right)^2$.

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

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

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

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 100 iterations. What do you observe? What does this plot tell you? [20pts]