CS2601 Linear and Convex Optimization
Homework 8

Due: 2021.11.26

For this assignment, you should submit a **single** pdf file as well as your source code (.py or .ipynb files). The pdf file should include all necessary figures, the outputs of your Python code, and your answers to the questions. Do NOT submit your figures in separate files. Your answers in any of the .py or .ipynb files will NOT be graded.

1. In this problem, you will use Newton's method to solve Problem 1 of Homework 7, i.e. (9.20) of Boyd and Vandenberghe,

$$\min_{x_1,x_2} f(x_1,x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1} \tag{1}$$

First implement the pure Newton's method in `newton.py`.

(a). Solve (1) numerically using your implementation of Newton's method. Use the initial point $\boldsymbol{x}_0 = (-1.5, 1)^T$. Report the solution and the number of iterations. Plot the trajectory of $\boldsymbol{x}_k$ and the gap $f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*)$.

(b). Repeat (a) for the initial point $\boldsymbol{x}_0 = (1.5, 1)^T$.

2. **Logistic regression.** First implement the damped Newton's method in `newton.py`. Then your implementation of damped Newton's method to solve Problem 3 of Homework 6,

$$\min_{\boldsymbol{w}} f(\boldsymbol{w}) = \sum_{i=1}^{m} \log(1 + e^{-y_i \boldsymbol{x}_i^T \boldsymbol{w}})$$

Recall

$$\nabla f(\boldsymbol{w}) = -\sum_{i=1}^{m} [1 - \sigma(y_i \boldsymbol{x}_i^T \boldsymbol{w})] y_i \boldsymbol{x}_i$$

(a). Show the Hessian of f is

$$\nabla^2 f(\boldsymbol{w}) = \sum_{i=1}^{m} \sigma'(y_i \boldsymbol{x}_i^T \boldsymbol{w}) \boldsymbol{x}_i \boldsymbol{x}_i^T$$

Note that here $\boldsymbol{x}_i$ is considered as a column vector, but in the implementation $\boldsymbol{x}_i$ is stored as the i-th row of the matrix $\boldsymbol{X}$, i.e. $\boldsymbol{X}^T = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m]$.

(b). Find the optimal $\boldsymbol{w}^*$ using your implementation of damped Newton's method. Use $\alpha = 0.1$ and $\beta = 0.7$ and initial point $\boldsymbol{w}_0 = (1, 1, 0)^T$. Report the solution, the number of iterations in the outer loop and the total number of iterations in the inner loop. Plot the gap $f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*)$ and the step sizes t_k.

Hint: You may consider using the broadcasting mechanism of python. Also note if $\boldsymbol{X}^T = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m)^T$ and $\mathbb{Z}^T = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m)$, then

$$\sum_{i=1}^{m} \boldsymbol{z}_i \boldsymbol{x}_i^T = [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m] \begin{bmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_m^T \end{bmatrix} = \mathbf{Z}^T \boldsymbol{X}$$

(c). What happens if you use (pure) Newton's method with $\boldsymbol{w}_0 = (1, 1, 0)^T$?

3. Consider the optimization problem $\min_x f(x)$, where $f : \mathbb{R} \to \mathbb{R}$ is given by $f(x) = (x - a)^4$, and $a \in \mathbb{R}$ is a constant.

(a). Find an explicit expression for the Newton step.

(b). Let x_k be the sequence of iterates generated by Newton's method. Let $y_k = |x_k - a|$ be the error between the k-th iterate and the optimal solution. Show that $y_{k+1} = \frac{2}{3} y_k$

(c). Conclude $|x_k - a|$ decays to zero exponentially, i.e. x_k converges exponentially to a.

Remark. Note that the convergence rate here is only exponential no matter how close x_0 is to $x^* = a$, while the rate given by the theorem in §7 is doubly exponential, which is much faster, at least when x_0 is close enough to x^*. This is because $(x - a)^4$ is not strongly convex, which does not satisfy the assumptions of the theorem.

4. Complete the implementation of the soft-thresholding operator and ISTA in `ista.py` for solving Lasso in penalized form,

$$f(\boldsymbol{w}) = \frac{1}{2} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{w}\|_1$$

We will explore the effect of λ on the number of zeros in the solution $\boldsymbol{w}^*$.

(a). Reproduce the result on slide 10 of §8, with

$$\boldsymbol{X} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} 3 \\ 2 \\ 2 \end{bmatrix}, \quad \lambda = 2$$

step size $t = 0.1$, and initial point $\boldsymbol{w}_0 = (0.5, 0.5)^T$. Report the solution and the number of iterations. Plot the trajectory of $\boldsymbol{w}_k$ and the gap $f(\boldsymbol{w}_k) - f(\boldsymbol{x}^*)$. Use the solution you find in place of $f(\boldsymbol{x}^*)$.

(b). Redo part (b) with $\lambda = 1$. Do you get zeros in $\boldsymbol{w}^*$?

(c). Redo part (b) with $\lambda = 6$. How many zeros do you get in $\boldsymbol{w}^*$?

Remark. There may be numerical errors in the $\boldsymbol{w}^*$ you obtain, but you can safely guess the exact $\boldsymbol{w}^*$ from the numerical solutions. If you want to verify your guess (you don't have to for this assignment), you can use the following condition: $\boldsymbol{w}$ is optimal iff

$$(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})_i + \lambda \operatorname{sgn}(w_i) = 0 \;\text{ if }\; w_i \neq 0 \qquad \text{and} \qquad -\lambda \leq (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})_i \leq \lambda \;\text{ if }\; w_i = 0$$