

## Convex Optimization Homework #2,

Due: Friday June 5, 2020, 9pm.

1. (100%) In this problem, we use the Newton's method to solve an unconstrained optimization problem. We wish to minimize

$$f(x) = \sum_{i=1}^m e^{a_i^T x - b_i}$$

where  $A \in \mathbf{R}^{m \times n}$  with  $a_i$  being the transpose of the  $i$ th row of  $A$ ,  $b \in \mathbf{R}^m$ , and the variable  $x \in \mathbf{R}^n$ . A special case of this is when we choose  $m = 3$ ,  $n = 2$ , and

$$A = \begin{bmatrix} 1 & 3 \\ 1 & -3 \\ -1 & 0 \end{bmatrix}, b = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}, \quad (1)$$

then it reduces to Eq. (9.20) in the textbook:

$$f(x) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}.$$

- (a) Find **dom**  $f$ . Show that **dom**  $f$  is a convex set.
- (b) Derive  $\nabla f(x)$  and  $\nabla^2 f(x)$  for any  $x \in \mathbf{dom} f$ .
- (c) Write a matlab m-file as a function which takes inputs  $A, b$ , and  $x$ , and evaluates the objective function at the point  $x$ , as well as the gradient and the Hessian.  
*Hint: the function can have a header that looks like the following.*  

```
function [f, g, H] = my_objective(x, A, b)
```
- (d) From this step on, write another m-file as the main function for your program to solve the unconstrained problem. Set  $A$  and  $b$  as in Eq. (1) and set the initial point  $x^{(0)} = 0$ .
- (e) Use the Newton step's formula:  $\Delta x_{\text{nt}} = -(\nabla^2 f(x))^{-1} \nabla f(x)$  and calculate the first Newton step  $\Delta x_{\text{nt}}^{(0)}$ .
- (f) Calculate the Newton decrement  $\lambda(x) = (-\nabla f(x)^T \Delta x_{\text{nt}})^{1/2}$ .
- (g) Perform backtracking line search along search direction using  $\beta = 0.7$  starting from  $t = 1$  until  $f(x + t\Delta x_{\text{nt}}^{(0)}) \leq f(x) - \alpha t \lambda(x)^2$ , where  $\alpha = 0.1$ .
- (h) Perform the update  $x^{(1)} = x^{(0)} + t\Delta x_{\text{nt}}^{(0)}$ . Determine whether  $\lambda^2/2 < \epsilon$  where  $\epsilon = 10^{-10}$ .
- (i) Repeat (e)-(h) for the next iteration. Wrap up these steps in a main loop. For the  $k$ th iteration, record the following items: (1)  $f(x^{(k)})$ , (2)  $\lambda(x^{(k)})$ , (3)  $t^{(k)}$ . Plot  $\lambda(x^{(k)})^2/2$  versus  $k$ .
- (j) Use the **cvx** toolbox to solve the problem. Compare the results (i.e., optimal value and optimal point) with those you obtained through Newton's method. Do they coincide with each other?
- (k) Set  $A = \begin{bmatrix} 1 & 2 \\ 1 & -2 \\ -1 & 0 \end{bmatrix}$ ,  $b = \begin{bmatrix} 0.2 \\ 0.1 \\ 0.3 \end{bmatrix}$ , and  $x^{(0)} = 0$ . Repeat (e)-(j).

### Homework submission guidelines:

- Submit your answer online as a set of three files: two m-files and a document file (in \*.pdf or \*.docx) that contains all answers (including plots) in this problem set.
- If you choose to do the homework in Python, then submit \*.py files instead of m-files.
- Submit your files online at the Ceiba website. No paper shall be handed in.
- Late submissions will be treated under the principle elaborated as follows.
  - (1) Homework received by 9pm, June 5 ( $t_1$ ) will be counted fully.
  - (2) Homework received after midnight (0am), June 6 ( $t_2$ ) will not be counted.
  - (3) Homework received between  $t_1$  and  $t_2$  will be counted with a discount rate

$$\frac{t_2 - t}{t_2 - t_1}$$

where  $t$  is the received time. Note that  $t_2 - t_1$  is three hours.

- **Plagiarism is strongly prohibited.** While discussions among classmates are allowed (and encouraged), you shall not ask anyone else to share his/her codes with you, nor should you attempt to share with anyone your codes. You should write every line of the code by yourself. If any part of your submission is found to be copied from someone else's submission, then **both of your homework submissions will be counted zero.**