CMO

Sheet 2 — E0230

Assignment (Due: 14 December 2020)

Instructions

- Answer all questions
- For numerical questions, answer without any leading or trailing spaces
- You can submit MS Teams Form **only once**
- If the value of answer is an **integer**, give answer in box provided as an integer alone **without any decimal point**. For example 0, 10
- Otherwise, if the value of answer is a **decimal number**, give answer in x.yy format **rounded to 2 decimal places**. For example 0.50, 1.50
- Follow this instruction unless otherwise specified in the question
- Late submissions will be penalised

1. *Quadratic minimization.* Consider the following minimization problem

$$5x^2 + 5y^2 - xy - 11x + 11y + 11$$

   (a) Let $[\hat{x}, \hat{y}]$ be a point satisfying the first order necessary conditions for a solution.

      (i) (1 point) $\hat{x} =$ ____________

      (ii) (1 point) $\hat{y} =$ ____________

   (b) (1 point) Is this point is a global minimum

   (c) (1 point) What would be the rate of convergence of steepest descent for this problem

   (d) (1 point) Starting at x=y=0, how many steepest descent iterations would it take (at-most) to reduce the function value to 10^{-11} (in integer format)

2. *Quadratic minimization.* Let $f : \mathbb{R}^4 \longrightarrow \mathbb{R}$ be twice continuously differentiable function. Assume that the hessian of f is positive definite and the largest absolute eigen value of the Hessian matrix of f at all points is bounded above by 25. We are given that $x_0 = [4, 0, -2, 1]^T$, $f(x_0) = 6$, and $\nabla f(x_0) = [8, 4, 4, 2]^T$. Using $2nd$ order Taylor series, find a quadratic function $g : \mathbb{R}^4 \longrightarrow \mathbb{R}$ such that $g(x_0) = f(x_0)$ and $f(x) \leq g(x), \forall x \in \mathbb{R}^4$. What is the minimum value of g (rounded to the nearest integer) (5 points)?

3. *Constant step-size.* Consider the function

$$f(x) = 3(x_1^2 + x_2^2) + 4x_1x_2 + 5x_1 + 6x_2 + 7$$

   where $x = [x_1, x_2]^T \in \mathrm{R}^2$. Suppose we use a fixed step-size gradient descent to find minimizer of f.

$$x^{k+1} = x^k - \alpha \nabla f(x^{(k)})$$

   Let $a < \alpha < b$ be the largest range of values of α for which the algorithm is globally convergent. Find a,b rounded to 2 decimal places in x.yy format.

   (a) (2 points) a = ____________

   (b) (3 points) b = ____________

4. *Constant step-size.* Consider the function $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$ given by

$$f(x) = \frac{3}{2}(x_1^2 + x_2^2) + (1 + a)x_1x_2 - (x_1 + x_2) + b$$

   where a and b are some unknown real valued parameters.

   (a) (2 points) Find the largest value of a (rounded to the nearest integer) for which unique global minimizer of f exists.

(b) (3 points) Consider the following algorithm

$$x^{(k+1)} = x^k - \frac{2}{5}\nabla f(x^{(k)})$$

Find largest value of a (rounded to the nearest integer) for which the above algorithm converges to the global minimizer of f for any initial point $x^{(0)}$.

5. *Steepest descent.* Write a subroutine (in Python) for implementing steepest descent using exact line search.

   (a) Use the given function `f5.pkl`, its gradient `grad_f5.pkl` and hessian `hess_f5.pkl`. The function in `f5.pkl` takes an argument $x \in \mathbb{R}^2$ as a python list of size 2 and returns a real number $f(x)$. Similarly, the function in `grad_f5.pkl` takes an argument x as a python list and returns $\nabla f(x)$ as a python list. The function in `hess_f5.pkl` does not take an argument and returns hessian matrix of $f(x)$. The functions can be loaded as `dill.loads(pickle.load(file_pointer))`, where `dill` and `pickle` are python libraries and `file_pointer` points to the pickle file to be read. Now, test your subroutine on $f(x)$ using initial condition $[0, 10]^T$. For the stopping criterion, use $||g^{(k)}||_2 \le \epsilon$ where $\epsilon = 10^{-6}$ and $||.||_2$ is 2-norm.

      (i) (2.5 points) Determine number of iterations required to satisfy above stopping condition?

      (ii) (2.5 points) What is value of objective function $f(x)$ at final point?

   (b) Now, test the subroutine for following quadratic problem $f(x) = \frac{1}{2}x^T Ax - b^T x, x \in \mathbb{R}^4$. For the stopping criterion, use $||g^{(k)}||_2 \le \epsilon$ where $\epsilon = 10^{-6}$ and $||.||_2$ is 2-norm.

   $$\text{with } A = \begin{pmatrix} 0.78 & -0.02 & -0.12 & -0.14 \\ -0.02 & 0.86 & -0.04 & 0.06 \\ -0.12 & -0.04 & 0.72 & -0.08 \\ -0.14 & 0.06 & -0.08 & 0.74 \end{pmatrix}, b = \begin{pmatrix} 0.76 \\ 0.08 \\ 1.12 \\ 0.68 \end{pmatrix}, x_0 = 0$$

      (i) (2.5 points) Determine number of iterations required to satisfy above stopping condition?

      (ii) (2.5 points) What is value of objective function $f(x)$ at final point?

   Give answers for (ii) rounded to 2 decimal places in x.yy format.

6. *In-exact line search.* Backtracking is a form of inexact line search in which a step size is determined at each step which satisfies the Armio-Goldstein condition. Given constants $\alpha, \beta \in (0, 1)$, at each step of the algorithm, if the current point is $x \in \mathbb{R}^d$, the direction of line search is chosen as $u = -\nabla f(x)$, and for determining the step size, an initial step size $t = 1$ is chosen and is repeatedly updated as $t \leftarrow \beta t$ until $f(x + tu) \le f(x) + \alpha t \nabla f(x)^T u$ and then x is updated as $x \leftarrow x + tu$. Once the update distance $||tu||_2$ for the point x

becomes less than ϵ during any epoch, the algorithm is stopped.
Use the given function `f6.pkl` and its gradient `grad_f6.pkl`. The function in `f6.pkl` takes an argument $x \in \mathbb{R}^4$ as a python list of size 4 and returns a real number $f(x)$. Similarly, the function in `grad_f6.pkl` takes an argument x as a python list and returns $\nabla f(x)$ as a python list. The functions can be loaded as
`dill.loads(pickle.load(file_pointer))`, where `dill` and `pickle` are python libraries and `file_pointer` points to the pickle file to be read.
Apply backtracking line search algorithm with initial point $[10, 100, 100, 10]$, $\alpha = 0.5$, $\beta = 0.5$ and $\epsilon = 10^{-7}$.

(a) Let $[x_1, x_2, x_3, x_4]$ the solution vector with each element rounded to 2 decimal places in x.yy format.

  (i) (1 point) $x_1 =$ ____________

  (ii) (1 point) $x_2 =$ ____________

  (iii) (1 point) $x_3 =$ ____________

  (iv) (1 point) $x_4 =$ ____________

(b) (3 points) Give the number of iterations it took to obtain the result.

(c) (i) (1.5 points) Give the least number of function calls to `f6` to obtain the result.

  (ii) (1.5 points) Give the least number of function calls to `grad_f6` used to obtain the result.

7. *One-dimensional search methods.* Consider the one-dimensional minimization problem

$$\min_{x \in [a,b]} \quad f(x) \tag{1}$$

We are given a, b and a subroutine 'foo'. The subroutine 'foo' returns the value function $f(x)$ for any $x \in [a, b]$. Let x^* be the unique minimizer for this problem. Given a tolerance value $\epsilon > 0$, we are interested in finding $\hat{x} \in [a, b]$ such that $|\hat{x} - x^*| \leq \epsilon$. Here is a pseudo-code to find $\hat{x}$.

- Initialize: $x_l = a$, $x_u = b$
- In loop:
  - $d = (x_u - x_l) * \rho$, $0 < \rho < 1$
  - $x_- = x_u - d$, $x_+ = x_l + d$
  - if $f(x_-) < f(x_+)$ then $x_u = x_+$ otherwise $x_l = x_-$
- Output: $0.5(x_l + x_u)$, tolerance $= 0.5(x_u - x_l)$, NSC = number of times the subroutine 'foo' was called.

Using the generic pseudo-code described above, we want you to implement two line search techniques for uni-modal functions. First is Golden section search (GS) where $\rho = \frac{\lambda}{(1+\lambda)}$ where $\lambda = 0.5(1+\sqrt{5})$. Now implement Golden section search as a function (in Python). Note that the loop part of GS continues until $tolerance \leq \epsilon$ is satisfied.

Second is Fibonacci search. In FS, k^{th} iteration uses $\rho = \frac{F_{N-k}}{F_{N-k+1}}$, where $F_0 = 1$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2} \geq 2$ is the Fibonacci sequence. Note that in FS, the loop will be executed $N-1$ times.

Implement the 'foo' subroutine such that 'foo(x)' returns $e^{-x} - cos(x)$ for $x \in [0,1]$. First call FS subroutine for $N = 20$, $N = 10$. Then call GS subroutine with ϵ returned by tolerance value of FS. Give answers for $\hat{x}$ and tolerance rounded to 3 decimal places in the form **x.yyy**.

(a) Run FS subroutine for $N = 20$ and report the following values

  (i) (1 point) $\hat{x} =$ ____________

  (ii) (1 point) $tolerance =$ ____________

  (iii) (0.5 points) $NSC =$ ____________

(b) Run GS subroutine using ϵ calculated in (a) and report the following values

  (i) (1 point) $\hat{x} =$ ____________

  (ii) (1 point) $N =$ ____________

  (iii) (0.5 points) $NSC =$ ____________

(c) Run FS subroutine for $N = 10$ and report the following values

  (i) (1 point) $\hat{x} =$ ____________

  (ii) (1 point) $tolerance =$ ____________

  (iii) (0.5 points) $NSC =$ ____________

(d) Run GS subroutine using ϵ calculated in (c) and report the following values

  (i) (1 point) $\hat{x} =$ ____________

  (ii) (1 point) $N =$ ____________

  (iii) (0.5 points) $NSC =$ ____________